The Design of a Computer-Based Pedagogy for Teaching Calculator Representations

Andrew Gordon Harrop

Thesis submitted in accordance with the requirements for the degree of Doctor of Philosophy

Computer-Based Learning Unit
School of Education
The University of Leeds

November 2001

The Candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others.

Abstract

This thesis demonstrates a novel method for improving the understanding of numerical structure in arithmetic through the use of computer-based multiple linked external representations. The system ENCAL exploits three representations: iconic, calculator and dataflow. Small-scale studies contributed to the design, and the results of a final evaluation study suggest that the approach can be usefully exploited in classroom mathematics education.

Cognitive science research has extended the concept of internal mental structures to include the interactions which take place between a person and the environment, such as technology-supported learning environments (Kozma, et al., 1996). The following thesis asserts that a computer-based learning environment facilitates the construction and use of mental models, particularly if one advocates the idea that cognition is viewed not as a purely mental process, but as a system which includes the individual, his/her social context, and the available cognitive tools - such as a computer (Dörfler, 1993). In addition, computer-based learning environments aid concrete to abstract thinking, because visually concrete objects can be linked to more formal and abstract mathematical representations (Kaput, 1989).

The mathematical problem solving ability of school children aged 12-13 years was assessed using the computer-based learning program ENCAL. The system helps children develop their concept of number and their skills with multiplication and addition with the help of a software calculator and some additional computer-based support. The aim of the evaluation was to ascertain the effectiveness of the software's three equivalent and linked representations: *iconic(concrete); datatree (intermediate)*; and *calculator (abstract)*; with regard to helping pupils solve text-based arithmetic problems. Two groups of mixed ability children were tested, one group had use of the intermediate datatree whereas the other group did not.

Overall, it was found that both high and low attainment pupils in the datatree group obtained a greater number of totally correct answers compared to the no-datatree group. Also, the low attainment pupils in the datatree group achieved notably more correct calculator answers for the most difficult question. In particular, those participants who did not have access to the datatree had operator and brackets (i.e. parentheses) problems in all three of the questions. However, the group who were able to use the datatree had no operator errors, and only two brackets errors with the most difficult question.

Acknowledgements

The author is indebted to Dr Paul Brna and Dr. Thomas Green for their significant help and advice throughout this research project. I would also like to express my grateful thanks to Mr Ken Tait, firstly for programming ENCAL and secondly for accommodating my frequent design changes. Thank you to the visiting teacher and pupils of Temple Moor High School for their help and co-operation without which this project would not have been possible.

I devote my heartfelt thanks to the staff of the Computer Based Learning Department for their tremendous help and support especially after becoming fully wheelchair bound.

Assistance was also provided by Disability Services during the latter stages of writing the thesis. The subsequent personal aid given by Rachel and Alice was considerable. Without this, completing the PhD would have been an enormous struggle.

Table of Contents

List of Figures

List of Tables

Chapter 1

The Research Problem

1.1 Introduction

The relationship between teaching and learning is of fundamental importance with regard to children's arithmetic understanding. This issue is highlighted by the research of Hiebert and Lefevre (1986) which was concerned with the association between the procedures children use during mathematical activities and the concepts they acquire. The correspondence between procedural activities and concept formation implies that the influence of representations (and their limitations – O'Reilly, 1999) used during teaching are likely to have a significant impact on understanding. The following thesis asserts that an appropriately designed computer-based learning environment can provide an optimal approach to teaching and learning from both educational and psychological perspectives. The chapter initially explains computer-based learning in the context of the current research. The next section outlines using the computer as a medium. Word problems are considered next with particular reference to the difficulty children have of translating information from a problem statement to the solution. Computation is then referred to, emphasising the notion of operation hierarchy (i.e. order of operations). The behaviour of calculators is addressed next, with reference to the confusion caused during evaluations by the use of different logic systems. Calculator use in schools is then briefly introduced. A summary of the research problem is then given. Finally, an outline of the thesis chapters is set out.

1.2 Computer-Based Learning

Three important aspects of computer-based learning are the *entities, metaphors,* and *icons* used in the external representations. A computer-based environment may make use of entities, metaphors, and icons at the interface to enhance learning because the use of computers: (a) provides opportunities to stimulate cognitive processes during learning such as problem solving, and (b) enable children to participate in learning by discovery (Ainsa, 1999).Greeno (1983) refers to *entities* in a representation as being: "the cognitive objects that the system [i.e. human] can reason about in a relatively direct way, and that are included continuously in the representation" (p. 227). The implication being that entities can facilitate learning since they are easily distinguished from other abstract relations, and therefore may be used as cues in the problem solving process. With regard to *metaphors*, Carroll and Mack (1985) point out that providing pupils with comparisons can help learning. In addition, they state that metaphors (particularly concrete metaphors) stimulate the construction of mental models and thus enhance learning. Finally, an *iconic* interface uses images to represent actual physical objects. For example, as in this study - bookcases, shelves, and books. This type of icon is a *representational* icon (Benbasat and Todd, 1993). A fundamental advantage of using this type of icon is: firstly, the primary task of problem solving is less interfered with because minimal effort is required to interact with such icons compared to text-based information; secondly, representational icons give contextual significance to a problem through the use of familiar objects which in turn assist with the *mapping* or *translation* of information from problem to solution; thirdly, the use of *metaphors* (which enhance understanding) is facilitated through the use of representational icons.

Another important issue from an educational point of view is a construction process called *reflective abstraction.* This process helps with the acquisition of more abstract mathematical concepts through the use of entities and metaphors. According to Piaget, reflective abstraction is a means by which students construct abstract structures as a result of a student reflecting on: (a) his or her own activities; and (b) the arguments used in social interaction (i.e. pupil-teacher, and pupil-pupil interaction).

Lehtinen and Repo (1996) believe that: (a) computer-based learning environments facilitate learning activities that are optimal for reflective abstraction; and (b) such learning activities could not be carried out in traditional classroom environments. In other words, Lehtinen and Repo (1996) state that a computer-based learning environment provides a technology-rich setting appropriate for activities which support the construction of mathematical concepts. What is also fundamental to a computer-based design for Lehtinen and Repo (1996) is the use of *multiple representations.* Lehtinen and Repo (1996) hypothesise that the use of multiple representations, and the movement between them stimulates reflective abstraction in students. Furthermore, the use of different external representations serves to inspire learners towards reflection and social interaction.

According to Hiebert and Lefevre (1986), "relationships at the reflective level are less tied to specific contexts. They are often created by recognising similar core features in pieces of information that are superficially different" (p. 5). For example, if a learner is required to add fractions, a connection at the reflective level is made if he or she is able to step back and reflect on the information previously provided for adding decimals, and from this reflection, appreciate the general idea that you always add things that are alike in some significant way. Thus, thinking at the reflective level means that a person is able to *generalise* the knowledge he/she already has to other mathematical situations. With regard to generalisation, Greeno (1983) points out that the effect of *mapping* in relation to arithmetic problems brings about a generalisation of procedures learnt by the pupils. Thus, if blocks and numerals are used for a subtraction problem, the entities which are referred to "are quantitative concepts for which both the numerals and the blocks provide symbolic representations" (p. 237), thus facilitating the acquisition of representational knowledge of subtraction.

Hiebert and Lefevre (1986) also point out that the subsequent development of *conceptual knowledge* is brought about by the construction of relationships (i.e. linking) between pieces of information. This linking of information can occur between: (a) pieces of information that are already stored in a person's memory; or (b) between existing knowledge in memory and knowledge which has been newly learnt. When previously unrelated items are suddenly seen as being related in some way, then there has been an increase in the *conceptual knowledge* of a learner (Hiebert, and Lefevre, 1986). In effect, learning has occurred by discovery (Bruner, 1961, in Hiebert and Lefevre, 1986). However, it is not enough to just have various representational tools available (e.g. computer-based representational tools). Rather, effective learning is facilitated by tools that are related to the new mathematical concepts to be learned and an awareness of the previous knowledge of the students. In the current thesis, the concepts are associated with the solving of arithmetic word problems and the previous knowledge would be that of pupils who have mathematical ability of around 10-13 years of age).

1.2.1 Problems With Current Software

With reference to computer-based tools, current computer software can create or solve arithmetic problems, however it appears that little or no computer-based research in the context of arithmetic word problems appears to address fundamental issues related to arithmetic understanding when solving problems. These are stated below:

- mappings to and from problem statements and arithmetic representations;
- conceptual and procedural understanding of multi-operator arithmetic computations using external representations;
- the use of an intermediate representation between the concrete and the abstract to facilitate understanding; and
- equivalent multiple linked representations to further enhance concrete, intermediate, and abstract understanding.

Thus, a way needs to be found which will improve software used for solving arithmetic word problems. Specifically, the software should facilitate the translation of information from the problem statement to the answer in the preferred sequence of concrete to abstract understanding. A child will therefore need computer-based support with mappings within and between the following: a problem statement, the subsequent arithmetic expression, and the computation of arithmetic expressions that involve order of operations, parentheses, and calculator use.

1.3 Using the Computer as a Medium

Repenning, Ioannidou, and Ambach (1998), make the distinction between the use of the computer as a *tool* and the computer as a *medium*. The difference is crucial as described below when considering the design of a learning environment.

When used as a tool, the computer is simply viewed as a mechanism for doing something, such as word processing, or calculating values. However, considering the computer as a medium increases its educational value since the computer is now viewed as a collection of tools which communicate something (e.g. objects used at the interface within a computer-based learning environment, the world-wide web, email, video, text, pictures, etc.). In other words, when used as an educational medium, computers diversify and facilitate the ways in which communication takes place (e.g. from teacher to pupils through the use of software and from pupils to pupils in subsequent social learning situations) and provide alternative communication experiences that enhance the learning experience. Conceptualising the computer as a medium also emphasises the *constructive* learning process. This is because such learning is characterised (Simons, 1993) as being:

active (i.e. a person interacts at the interface in a meaningful way)

cumulative (i.e. new learning is constructed based on prior knowledge) and

goal oriented (i.e. learning is more likely to occur if a learner is aware of the goal).

Although the use of a computer implies a one-to-one (sit alone) interaction between the learner and the computer, the theory of constructivism suggests that teaching and learning activities using a computer will promote reflective and meaningful thinking. This will involve subsequent social interaction both between pupils and pupils and between teachers and pupils. Hoyles, Sutherland, and Healy (1991), point out that such social interaction provides cognitive scaffolding during problem solving tasks.

The social nature of teaching and learning is further emphasised by the cognitive science theory of *distributed cognition*. Dörfler (1993) points out that cognition should be viewed as being distributed over a whole system rather than just being confined to an individual. The system being comprised of: (a) the individual; (b) his/her social context; and (c) available cognitive means (such as, other people, writing, numerals, representational systems, calculation tools like the abacus, slide rule, calculator, or computer). Therefore, cognitive development is not viewed as a purely mental process, but cognition is seen as being distributed over the tools that facilitate a person's thinking and problem solving.

Similarly, the active nature of the constructive learning process and the social/cultural aspect of learning is fundamental to *activity theory*. The theory asserts that consciousness is not confined just to cognitive functions, such as decision making and memory. Instead, consciousness is associated with social interaction – that is, what people do and the artefacts people interact with (Nardi, 1996). Thus, activity theorists argue that consciousness is constructed during development as a result of a person's activity – that is, there is an underlying unity between a person's activity and his/her consciousness.

Constructivism, distributed cognition, and activity theory imply that learning is not simply a passive or a purely externally directed process, but is active, constructive, and self-directed in which a person builds up internal knowledge representations based on his or her learning experiences (Vermunt, 1998). Thus, learning is more under the control of the student, as opposed to being externally driven as in traditional instructional teaching – where the emphasis is on transferring knowledge from an external agent to the learner. A study by Marton and Säljö (1984) found that instructional teaching did not induce a meaningful, deep approach to learning, because students were more preoccupied with answering the set questions, as opposed to engaging in a more in-depth approach to learning. That is, learning which would help a person construct a new knowledge representation to add to his or her existing knowledge - such as the understanding of a concept. Research by Vermunt (1998) supports this finding. In particular, the results indicate that high-quality learning is best achieved by transferring control over the learning process from teachers to pupils such that teaching influences more of a constructivist mode of learning. In other words, external regulation (such as directions for learning provided by teachers) has little influence on students' processing strategies during learning, but rather it is learners themselves who regulate their learning process.

Consequently, the rationale underlying the research in this thesis is as follows. Firstly, the computer is seen as an integrated constructive educational *environment* that facilitates the teaching and learning of mathematics (i.e. elementary arithmetic word problems) in the classroom setting. More specifically, the computer environment is viewed as facilitating the construction of pupils' learning processes as described below.

1. By providing: primarily, a social learning environment based on self-regulated exploration, reflection, and collaborative learning; and secondarily, external regulation (such as problem setting and feedback) from the teacher to support learning.

2. Through the process of user interaction, the computing environment is seen as enabling (a) teaching and learning to be closely related in terms of being social, constructive, active, and meaningful (Shuell, 1996), (b) teaching and learning to occur with understanding (i.e. to facilitate mathematical mental representations to become part of a person's existing knowledge network), (c) concrete to abstract correspondence to take place, since this process is a fundamental prerequisite to the teaching, learning and understanding of mathematical concepts and procedures regarding the solving of arithmetic word problems, and (d) the perception by learners of constructs/representations of the computer environment since these will receive individual meaning by users.

3. The influence of the computer-based learning environment is seen as leaving a "lasting cognitive residue" (Salomon, 1992) and as a result will serve to enhance both teaching and learning.

1.3.1 The use of Icons

A computer-based environment may make use of icons at the interface to enhance learning. This is because an iconic interface can uses images to represent actual physical objects. For example, the images might represent bookcases, shelves, and books. This type of icon is a *representational* icon (Benbasat and Todd, 1993). There are two fundamental advantages of using this type of icon. Firstly, the primary task of problem solving suffers from less interference because minimal effort is required to interact with such icons compared to text-based information. Secondly, representational icons give contextual significance to a problem through the use of familiar objects (e.g., book and shelf icons) that in turn assist with the *mapping* or *translation* of information from problem to solution. Thirdly, the use of *metaphors* (e.g., the use of rectangles to represent brackets) is facilitated through the use of representational icons.

1.4 Word Problems

A recurring fundamental difficulty encountered during the solving of arithmetic word problems lies in the translation of information from the problem text to mathematical language as pointed out by Nesher (1988).

In this thesis, emphasis is placed on the arithmetic required for solving a problem rather than with the textual interpretation. The type of problems considered in the following research are referred to as *repeated addition.* The reasons for focusing on this type of problem are firstly multiplication and addition operations are used which involve mappings between entities, secondly most confusion usually occurs when mixed operations are used which are not inverses of each other (e.g. + and x) as pointed out by Ecker (1989) and thirdly, it is considered to be the easiest type of "multiplicative" (Nesher, 1988, p. 21) word problem in terms of comprehension.

This type of word problem is characterised by the use of three propositions as follows:

There are n_1 x's (e.g. shelves) for which there are y's (e.g. books).

For all x's if there are y's, then there are exactly n_2 y's (books per shelf). This describes a mapping between the shelves and books.

The final proposition asks how many y's (books) are there for all the x's (shelves).

Movement from a problem statement of the type described above to a mathematical solution is a complex process and involves concrete to abstract understanding. The relationship between concrete and abstract in mathematics education can be difficult for teachers to convey and pupils to understand. In particular, De Corte and Verschaffel (1987) point out that a major difficulty during problem solving is the construction of a suitable problem representation. Greer (1992) points out that the competent solving of word problems requires a child to: (a) translate from the natural language representation of a problem to the mathematical language representation; and (b) use an "intermediate representation" (p. 285) to help with the construction of the appropriate mathematical formulation. Nesher (1980) refers to findings which suggest that students tend to move directly from a problem statement to the formulation of mathematical expressions based on "surface clues" or trial and error, without first considering some form of intermediate representation to assist in the problem solving process. An appropriately designed computer-based learning environment could be used to assist in the competent solving of word problems by providing an intermediate representation to conceptually "bridge" the concrete and the abstract.

Although much computer-based software has been directed at helping children plan and solve arithmetic word problems (e.g., SEMCALC, Schwarz, 1982; EDUCE, LeBlanc and Russell, 1989; ICE, Kaput, 1989; TAPS, Derry, Hawkes, Diefenbach, and Kegelmann, 1993; ANIMATE, Nathan and Resnick, 1993; PLANNER, Schwarz, Nathan, and Resnick, 1996), the use of specific intermediate representations to assist with the translation from natural language to mathematical expressions tends not to be addressed. However, some software has attempted to support concrete to abstract thinking or novice to expert understanding using *multiple representations*. For example, in the context of science, Kozma, Russell, Jones, Marx, and Davies (1996), used such representations to facilitate the understanding of chemical equilibrium. In the context of arithmetic word problem solving in primary school mathematics, Kaput (1989) developed a system to support the concrete to abstract thinking required during the multiplicative reasoning of problems such as: "How many apples will be needed altogether if four children are to get three apples each?" (Kaput, 1989, p. 38).

1.5 Computation

A further aspect of problem solving is the fact that once information has been elicited from the semantics of a word problem, a child will need to *compute* the answer. He or she will need to take into account: order of operations, parentheses, and calculator behaviour. These aspects of computation are referred to below.

To correctly compute an arithmetic function that results from a particular word problem, it is often first necessary to consider the order in which operations are to be carried out. Ecker (1989) highlights the computation problems of order of operations. For example, when considering 1 + 2 x 3, he states do you first add 1 and 2, and then multiply the result by 3, or do you multiply first, and then add? As mentioned previously, Ecker (1989) points out that most confusion usually occurs when mixed operations are used which are not inverses of each other (e.g. + and x). This ambiguity has been resolved without the use of parentheses by using an order of operations hierarchy. Priority is given to operators in the following hierarchy: division/multiplication, addition/subtraction. For example:

$8 - 2 \times 3 + 8 \div 4 =$

$8 - 6 + 2 =$

$2 + 2 =$

4

Where expressions use parentheses, then operations in the parentheses have top priority. When parentheses occur inside of parentheses, the innermost expressions are evaluated first. For example:

$24 \div (4 \times (2 + 1)) =$

$24 \div (4 \times 3) =$

$24 \div 12 = 2$

Significant computation errors made by pupils occur through lack of understanding of parentheses (Demana and Osborne, 1988). For example, (3 + 5) x 4 is often thought to be equivalent to 3 + 5 x 4. Four-function (arithmetic) calculators contribute to such misunderstandings, and thus "fail to give correct values of mathematical expressions needed for appropriate pre-algebra experiences" (Demana and Osborne, 1988, p. 3). Typical problems associated with calculator use are outlined below.

1.6 The Behaviour of Calculators

Children use calculators in both primary and secondary schools for the computation of arithmetic functions. An important aspect of calculator use is that it structures arithmetic expressions in specific ways dependent on the logic system used, and consequently the behaviour of a calculator is often at variance with the computational procedures pupils have been taught. For example, the distributive law states that (7 x 10) + (6 x 7) = 70 + 42 = 112. However, if a four-function (arithmetic) calculator is used to compute this expression, then children who tend to have a strong left-to-right bias and who perhaps do not understand order of operations and the meaning of parentheses, could type in the data from left to right and get the answer of 532. Therefore, calculators could produce misinterpretations in children's understanding of arithmetic. Wiebe (1989) points out that most pupils need help when using calculators, "especially if they are using them with problems involving more than one operation..." (p. 36). He states that different calculators and computer software tools use different internal algorithms for computing answers and thus different answers may be given to the same problem. For example, the sequence 4 + 5 x 3 =, if entered into a four-function (arithmetic) calculator and an algebraic notation (scientific) calculator will give the answers 27, and 19 respectively. As Wiebe (1989) points out, pupils are astonished that a calculator may produce incorrect answers - they usually assume that they have entered data incorrectly. The evaluation of expressions is a persistent problem. Even at college level, calculator users have trouble with premature commitment (Green and Petre, 1996), in that they tend to work from left to right rather than manipulating an expression (Mayer and Bayman, 1981).

Ruthven and Chaplin (1997) looked at the part played by calculators in children's numerical learning who were aged 10/11 years. One aspect of the study analysed how pupils approached number problems with and without the use of calculators. The research showed that regardless of current teaching policies, outcomes of calculator use could be much influenced by the manner of introducing and using calculators in the classroom. The successful solution of arithmetic expressions requires mapping, and the study also highlighted the nature of the difficulties pupils had with mapping features of a word problem to and from the actions performed on a calculator. In particular, the findings showed that whether pupils used calculators or not, they were poor at representing and using intermediate results.

A fundamental problem is the fact that calculators look concrete, but they do not give perceptual representations to the underlying abstractions (i.e. the symbols and algebra used in expressions). Calculator teaching therefore needs a concrete graphical representation in order to support the conversion from concrete to abstract by using an intermediate view that gives a perceptual representation to the abstractions. Consequently, the use of intermediate results during calculations needs to be promoted (Ruthven and Chaplin, 1997), because calculators are poor at showing the intermediate stages of computations. Moreover, the step from concrete arithmetic understanding to the abstract order of operations used in calculations is probably too large. This further justifies the need for an intermediate representation to be adopted to address these issues.

1.7 Calculator Use in Schools

The use of calculators in schools is a controversial topic because they are widespread, powerful, and yet problematic.

Despite all the pedagogical debates, calculators are here to stay; and it would seem that teaching their use is desirable, as suggested by Fuson (1992). Still more relevant with regard to this thesis, is that they offer an excellent opportunity to investigate ways to teach the understanding of arithmetic structure. In particular, the fundamental importance of acquiring appropriate computational procedures as a basis for understanding the number system as put forward by Ohlsson (1987) and Bell, Costello, and Küchemann (1983), and the order of operations using calculators (Ecker, 1989; Wiebe, 1989).

1.8 Summary

Much computer-based research has been devoted to the comprehending, planning, and solving of elementary word problems. However, once information has been gleaned from a problem, abstract arithmetic understanding is a fundamental prerequisite for the subsequent formulation and evaluation of an arithmetic expression. A neglected issue appears to be finding a way which will improve not only concrete to abstract understanding but the mappings which are made between problem statements, arithmetic expressions and computational procedures (particularly those which involve the ordering of operations and the use of calculators). The requirements for optimal teaching and learning are the use of computer-based representations, (one of which should be an *intermediate* representation to facilitate such understanding) and a computer-based environment based on the constructivist perspective, such that the environment only becomes meaningful through the process of user-computer interface interaction.

1.9 Organisation of the Remainder of the Thesis

The remainder of the thesis will be organised as follows. Chapter Two considers some of the computer programs that have been designed and used for teaching arithmetic, and the limitations posed by such software. Based on the shortcomings of current software, this chapter addresses the pedagogical and computer-based usability requirements which will be needed for teaching and understanding abstract arithmetic representations, specifically those associated with calculator use. Chapter Three states the computer-based pedagogical solution that will be required for the teaching and learning of abstract arithmetic representations during elementary problem solving. Two design specifications (i.e. the internal/external representations and the iconsworld language) are addressed from a conceptual perspective in Chapter 4. Storyboard version

1 (i.e. design 1) for the computer-based learning system ENCAL is outlined in Chapter Five. The results of the usability pilot evaluation are also given for design 1. Design changes in the form of storyboard 2 (i.e. design 2) are then set out and are based on the pilot test results. Chapter Six recapitulates the theoretical grounds behind ENCAL. The final evaluation and the results are presented in Chapter Seven. Analysis of errors raised during the evaluation of the ENCAL system are addressed in Chapter 8. An overall discussion followed by the conclusions are then given in Chapter 9. Chapter 10 states the contributions made during this study, and future work which could be carried out to further improve ENCAL. The references are then listed. Finally, the appendices follow.

Chapter 2
Pedagogical and Usability Considerations

2.1 Introduction

What is of concern are the difficulties pupils (around age 10-13 years) encounter with abstract arithmetic representations (i.e. notation used with calculators), and how such difficulties may be overcome in a teaching environment. Section 2.2 initially outlines the theories underlying meaningful learning, constructivism, mental models, and considerations of representation, all of which have significant implications for the learning of arithmetic, particularly with the understanding of abstract representations. Based on these theories, the potential problems encountered in understanding abstract representations - specifically calculator representations are then addressed. This is followed by the proposed pedagogical requirements needed to achieve an understanding of calculator representations. Section 2.3 reviews computer software used for teaching arithmetic, and the limitations such software poses for meeting the pedagogical requirements needed for teaching calculator representations. A possible solution to the inadequacies of current software is put forward in Section 2.4 through the use of multiple, equivalent, linked representations (MELRs). Usability requirements underlying the proposed teaching system are outlined in Section 2.5. Finally, Section 2.6 summarises the pedagogical and usability requirements.

2.2 Learning Using Representations

2.2.1 Pedagogy Required for Teaching/learning Arithmetic Representations

A fundamental aspect of learning arithmetic is understanding and becoming proficient in abstract representational systems that convey concepts, such as algebra. A way of helping learners to understand, particularly in the primary school, is to make connections between familiar concrete situations and more abstract symbolic concepts. This may be achieved through the use of objects that can be manipulated (e.g. base-10 blocks, sticks, counters, and computer graphics). Piagetian analysis indicates that for young children, the use of concrete manipulatives is important for the eventual development of formal operations (Shuard, Walsh, Goodwin, and Worcester, 1991). Children around 7-12 years of age have the ability to think logically (i.e. like adults) if their thinking is guided by contact with real or familiar objects and situations. The physical activity of say moving blocks, eventually leads to similar actions being carried out entirely in the imagination, and so at around 11-12 years of age, mental activities come to dominate and take the form of mental images which are moved around in the mind. It is the construction of mental images that are fundamental to the development of logical thinking and thinking in terms of concepts.

Mayer and Wittrock (1996) refer to this learning process as being structure-based, meaningful, and active. Meaningful learning is particularly appropriate to abstract concept acquisition, since this is: (a) concerned more with understanding than just a change in procedures; and (b) influenced by situations and domains as opposed to being independent from them (Shuell, 1992). A related theory is Constructivism which asserts that both the active involvement in a situation by individuals, and the situation itself, affect cognitive growth (i.e. learning). Such constructive activity is based on Piaget's notion of assimilation and accommodation. That is, new procedures may be assimilated prior to learning, but meaningful learning (i.e. the understanding of concepts) occurs only after the accommodation of new schemata (Steffe, 1988). Thus, a new concept may be added to concepts already formed (i.e. constructed) as long as the new concept is accommodated to the existing concept network through understanding. If understanding does not take place then an individual will simply assimilate new information and cognitive growth will not occur.

Cognitive psychology has grounded this analysis in the concept of mental models. For example, the theory of language understanding put forward by Johnson-Laird (1983) suggests that children construct concrete mental models which correspond to the entities (e.g. people, objects, events) which the language is about (as someone would do when listening to a story). A child then manipulates and mentally transforms the mental model, and this enables inferences concerning the language to be drawn.

Greeno (1991a) proposed an idea of mental models related to number sense similar to Johnson-Laird's (1983) language theory. Greeno (1991a) suggests that number sense is a form of cognitive expertise - in other words, it is the ability of a person to construct and reason with mental models. Thus, understanding the language of mathematics (e.g. a word problem) depends on learners developing the ability to construct mathematical *situations* which include the concepts that the language is putting forward. Greeno (1991a) refers to this as *situated cognition.* His underlying assumptions concerning learning are that the capabilities which people have with regard to number sense involve more than just facts and procedures and that the activity of understanding and reasoning ultimately becomes internal (i.e. implicit) through the use of mental models.

The mental model theories above imply that when confronted with an abstract proposition, children need not think logically because they construct mental models of a situation by relating a proposition to the concrete or real world. Furthermore, since communication necessitates the use of external representations (e.g. with objects, symbols, and language), it may be assumed that internal representations (mental models) are influenced and constrained by external situations (situated cognition) and connections between internal representations may be achieved as a result of external activity, thus facilitating the construction of knowledge networks (Hiebert and Carpenter, 1992). Consequently, there should be decreased dependency on the use of physical aids (e.g. arithmetic blocks) to facilitate thinking. Thus, when attempting to solve a problem in the conceptual domain of numbers, a person's reasoning will be guided by his/her internalised mental models as formed through previous interactions of external concrete situations.

A fundamental aspect associated with both internal mental states of mind and external concrete situations is the pedagogical considerations of representation. These will be addressed next.

2.2.2 Pedagogical Considerations of Representation

Understanding mathematics is associated with the way information is presented and structured. The presentation of information should enable connections between ideas, facts, and procedures to be made. Cognitive science suggests that subsequent understanding will occur once a mental representation of a particular mathematical concept has become linked to a person's existing network of representations (i.e. from a psychological perspective, information has been accommodated).

With regard to representations and understanding, Hiebert and Carpenter (1992) build on two assumptions from cognitive science research by suggesting the following. Firstly, an internal representation is influenced by the represented external situation. Thus, connections between internal representations are influenced by connections between corresponding external representations, and so external mathematical representations influence internal mathematical representations. Secondly, internal representations can be connected. Similarly, Anderson (1993) states that the mind is a reflection of the external environment. However, Zhang (1997) goes further by arguing that external representation based problem solving (e.g. arithmetic multiplication) is constrained both by the environment and by the mind of an individual. The fundamental assumption being that external representations do not have to be re-represented as internal representations in order for problem solving to be carried out. Zhang suggests that external representations can activate perceptual operations and provide perceptual information which may be used in conjunction with cognitive operations, such as existing internal representations (e.g. information recalled from memory).

Such theories are particularly useful when considering the design of learning environments in the field of arithmetic problem solving. Goldin (1987) describes the goal of mathematics education as being able to foster the development of cognitive (i.e. internal) representational systems. Consequently, Goldin points out that a teaching system needs to foster maximal development of pupils' internal representations using external representations which facilitate transfer of learning. The above assumptions from cognitive science suggest that having the ability to select an action in one external representation which can then be translated to other connected external representations would be a powerful pedagogical tool for activating and providing perceptual information and also representing ideas mentally. In this respect, the computer is extremely useful, because several different external representations can both be linked and be structurally equivalent in order to facilitate the translation of actions carried out by pupils.

The distinction between internal and external representations helps understand why a particular technology may be better suited to the teaching and learning of arithmetic representations which involve the use of a calculator. Internal representations are the mental images people formulate in their minds which correspond to reality, whereas external representations are actual commodities which people can physically see and/or manipulate to depict reality, such as: symbols (e.g. algebra, diagrams, pictures); and real objects (e.g. arithmetic blocks, Cuisenaire rods, etc.). A representation may therefore be considered as being comprised of the three components: mental images, symbols, and real objects (Janvier, 1987).

Several different (i.e. multiple) types of external representation may be used to depict the same abstract concept that in turn could help to promote understanding. For example, different types of external representation typically used in classrooms include textbooks, writing/diagrams on white/black boards and manipulable objects. However, Dufour-Janvier, Bednarz, and Belanger (1987), point out that multiple external representations will only be useful if a child understands them. Where this occurs, a learner will also be expected to find one representation which best enables a given task to be completed, reject a representation because it is less effective than others in a given problem situation and move from one representation to another.

The movement between representations is problematic when considering how a solution is arrived at from a given problem statement. For example, Lesh, Post, and Behr (1987), found that pupils have "translation" (p. 36) difficulties associated with the re-representation of initial word problem information into ways of describing, illustrating, and manipulating ideas which may then be used for the solving of a problem. They point out that these difficulties arise not only within the context of word problems, but also with the translation to subsequent pencil and paper computations. Both the translation of information from word problem statements and the subsequent translation of information to computations are seen as significant factors in influencing mathematical learning. The translation difficulties highlighted by Lesh *et al.* will by implication also be apparent when considering the use of abstract calculator representations to facilitate computations.

The processes of mathematical thinking required to overcome such problems are based on complex relationships between the external representations encountered during learning, and internal mental processes (De Corte, Greer, and Verschaffel, 1996). A theoretical model that classifies the unobservable thinking behaviour (i.e. internal representations) taking place in individuals is Goldin's (1992b) model of internal representational systems. The model depicts complex processes of interaction between the following five internal representational systems: verbal/syntactic; heuristic (e.g. planning, and monitoring); formal notation (i.e. symbolic); affective; and imagistic (i.e. visual, spatial, auditory, tactile).

Conventional teaching tends to place an emphasis on verbally mediated thinking during mathematics teaching (e.g. through explanations), and the use of formal types of notation (e.g. algebra). Consequently, De Corte, *et al.* (1996) point out that Goldin's (1992b) model of mathematical thinking goes far beyond the presumed influence of verbal and formally written mathematics. Goldin's model implies that mathematical understanding (and thus teaching) needs to address the influence on learning of non-verbal imagery (i.e. visually mediated thinking), as well as the use of verbal and formal notational systems. This implication supports Paivio's (1986) dual-coding theory.

Having addressed relevant issues of learning and representation, the following section considers the pedagogical problems of teaching and learning using calculator representations.

2.2.3 Pedagogical Problems With Using Calculator Representations for Teaching/Learning Arithmetic

Previous research asserts that the construction of concepts should precede the use of skills (Kaput, 1987; Hiebert, 1988). However, calculator usage typically promotes skills before arithmetic understanding. This is evident when one considers the theories underlying: meaningful learning, constructivism, mental models, and internal/external representations, all of which suggest that calculators will be of little educational value unless pupils understand the formal/abstract representations used.

Children, particularly at primary school level, cannot easily relate abstract data on a calculator display to concrete or real-life situations, and this will serve to impede mental model formation and thus learning. More specifically, because internal representations are influenced and constrained by external situations and, since the external representations of calculators are themselves abstract, it may be assumed that calculators constrain the development of internal representations and thus conceptual networks. In addition, the symbols viewed on a calculator display refer to abstract entities which are likely to be absent from pupils' cognitive structures (Greeno, 1991), and if this is the case, new information may be assimilated but not accommodated. Thus, if learning using calculators is to take place, it is important that number concepts are represented internally in a way that promotes understanding.

Unfortunately, conventional calculator representations do not lend themselves to mental model formation and thus conceptual or procedural understanding, for several reasons. Firstly, although calculators look concrete, they do not give perceptual representations to underlying abstractions (e.g. mappings between calculation steps, and evaluation sequences). In other words, data which is input remains abstract (i.e. in a symbolised format) which itself is unlikely to facilitate accommodation of knowledge and conceptual understanding. Secondly, calculators can cause confusion about procedural understanding (such as order of operations) due to the logic systems used to implement the calculation. Thirdly, calculators do not show intermediate stages of computations that could serve to support abstract understanding, and thus help with the construction of mental models and the accommodation of knowledge networks. For example, the reading of a word problem to the entering of data into a calculator is probably too large a step for the understanding of the entities involved, the relationships between the entities and the evaluation sequence of an arithmetic expression. Fourthly, calculators do not facilitate planning, in particular the editing and reorganisation of data.

The potential pedagogical difficulties with calculator usage affect both teaching and learning, and so the next section considers the subsequent pedagogical requirements which will be needed to overcome such problems.

2.2.4 Pedagogical Requirements for Understanding Calculator Representations

What pedagogical requirements will be needed to facilitate the teaching and understanding of calculator representations? This is the fundamental question to be addressed. The specific pedagogical requirements will need to include the order in which information from a problem statement is both used and represented.

The logical order in which a user would need to consider information depicted in a problem statement when he/she uses calculator representations during problem solving is shown in points 1 to 7 below. Of course, users may depart from this order - e.g. revisiting earlier steps to remind themselves.

1. Identify the entities from a problem, and then represent them using a familiar *concrete* format

2. Identify the relationship between one entity from a set, with an entity from a different set (i.e. identify one entity per other entity - e.g. one book per shelf).

3. Identify a class of entities (e.g. several books) using the concrete format, and then assign a number to state how many (e.g. 4) using an intermediate representation which lies between the concrete format, and the abstract symbols used with the calculator.

4. Identify the relationships between the entities using the intermediate representation.

5. Convert the intermediate representation to the *abstract* notation used with the calculator.

6. Carry out computations with the calculator, and use the intermediate representation to assist with order of operations.

7. Scaffold the concrete format away over time.

The above pedagogical requirements should provide pupils with a teaching/learning situation that will facilitate arithmetic understanding when calculator representations are used. However, a remaining issue concerns the technology that can best meet these requirements. This is considered below.

2.3 What Technology Can Be Used to Achieve the Above Pedagogical Requirements?

Computer-based representations offer a preferable means of meeting the above pedagogical requirements based on the psychological and educational theories referred to earlier (see Section 2.2), for three fundamental reasons. Firstly, a specific program architecture may be built, and a suitable interface designed, to optimise teaching and learning. Secondly, representations may be explored to enhance learning using visualisation, metaphors, and direct manipulation. Thirdly, representations can be exploited via the computer interface to facilitate: (a) the linking of visually concrete objects to abstract arithmetic expressions; and (b) the construction and use of mental models to promote understanding (Dörfler, 1993).

Having stated that computer-based technology offers an optimal means of teaching calculator representations, it will be argued that external representations used in current educational software do not lend themselves to the teaching and understanding of calculator representations for the solving of arithmetic expressions based on initial problem statements. The following section outlines the pedagogical problems with such software for teaching calculator representations in the domain of arithmetic.

2.4 Computer-Based Technology and its Pedagogical Problems

2.4.1 Software Which Focuses on Arithmetic Computations

Computer assisted Learning (CAL) programs tend to focus on the arithmetic computations required for the solving of numeric expressions. For example, a typical drill and practice (i.e. early CAL) program presents users with expressions such as: 9 * 5 = __ (as described in Solomon, 1986). Feedback messages, such as TRY AGAIN are shown if an incorrect answer is given. Some CAL programs provide more sophisticated feedback, such as: *Good*, and *Now it's right*, in response to information input by users (e.g. Tait, Hartley, and Anderson, 1973). The CAL programs MUMATH (as described in Heid, 1990) and TOONTALK ARITHMETIC (Kahn, 1996), allow algebraic and simple arithmetic expressions respectively to be input and evaluated, but unlike their predecessors do not give written feedback regarding answers, since these systems perform computations automatically.

BUGGY (Brown and Burton, 1978), is an intelligent tutoring system (ITS) which goes further by diagnosing arithmetic performance. The program enables users to establish the procedural mistakes that could be made during numeric problem solving. Arithmetic expressions are presented vertically to make a pupil's working environment as close as possible to the pencil and paper way of writing arithmetic. For example:

1 8

+ 6

2 3

The computer plays the part of the pupil, and produces feedback in the form of incorrect answers to arithmetic problems, as shown above. The user attempts to identify the procedural error or bug that resulted in the incorrect answer. Once the bug has been discovered, the computer asks the user to describe it. The learner is then given five problems to solve, but he/she has to answer them incorrectly by utilising the procedural bug that has just been found.

ITSs that provide more detailed feedback of computation procedures are those of Attisha and Yazdani (1983), and Attisha and Yazdani (1984). Users attempt to answer problems correctly, and the system feedback informs pupils whether answers are correct or incorrect and what the possible causes of error are. As with BUGGY above, the problems are presented in traditional column format. An example of the computer output for a subtraction problem (refer to Attisha and Yazdani, 1983) is shown below.

ENTER TOP NUMBER 730

ENTER BOTTOM NUMBER 185

730 IF NUMBERS CORRECT

- 185 TYPE Y ELSE TYPE N

Y

TYPE YOUR ANSWER

655

YOUR ANSWER IS WRONG

POSSIBLE CAUSES OF ERROR:

YOU DID NOT BORROW, IN EACH COLUMN YOU SUBTRACTED THE SMALLER DIGIT FROM THE LARGER ONE.

WOULD YOU LIKE TO TRY AGAIN TYPE Y IF YES

Y

730 THESE ARE YOUR

- 185 NUMBERS AGAIN

ENTER ANOTHER ANSWER

545

YOUR ANSWER NOW IS CORRECT

More recent ITSs have concentrated on fraction computational procedures (e.g. the Fraction Intelligent Tutoring System, FITS - Nwana, 1993; and FRACT-2, referred to in Dumont, 1993). With the former system, questions (e.g. ¼ + ½) are presented, and computer feedback is provided through the use of hints (e.g. *What operation do you want to perform?*), or by prompting users to choose one of several options (e.g. *Sum the whole numbers*). The dialogue interaction between pupil and tutor continues until the answer is arrived at. The feedback used in the latter system provides remedial information that explains how an incorrect answer could have been arrived at by a user. Thus, if the wrong answer of 7/9 was given to the problem: 31/18 + 11/4, then the system would suggest that the incorrect numerator answer occurred by adding the numerators and dividing by 6. Similar mal-rules for incorrect denominator answers are also provided. The idea underlying the system is that similar errors may be avoided in the future.

2.4.2 Pedagogical Problems with the Above Software when Representations needed for Calculator Understanding are used to Solve Arithmetic Problems

Both the CAL and ITS software described above concentrate solely on arithmetic computation procedures. In addition, the systems either do not give users control over calculation procedures (i.e. the computer arrives at an answer without articulating its behaviour - "black-box" approach), or they emphasise traditional pencil and paper methods of computation as opposed to multi-operator calculator calculations. Such systems will not promote conceptual understanding largely because users are constrained to text-based representations that depict abstract mathematical symbols. Therefore, understanding a particular computation procedure (e.g. how to subtract 2-digit numbers from each other, or how to add two fractions) is not a sufficient requirement for solving arithmetic problems when: (a) they are presented in statements; and (b) a calculator is to be used for evaluations.

Although diagnosis of particular computation procedures is valuable, fundamental aspects of the pedagogical requirements outlined in Section 2.2 (points 1-7) are not met by this type of software. These are: identifying and representing entities from a problem statement in a concrete way (point 1); using an intermediate representation which lies between concrete graphics and abstract notation (point 3); identifying relationships between the entities (point 4); using an intermediate representation to facilitate the transition from concrete forms to abstract symbols (point 5); and using the calculator and intermediate representations to help with order of operations understanding during calculator computations (point 6).

Since the types of software described in this section neglect the above pedagogical requirements, such systems will not be considered further. Instead, the following sections will overview software that meets the pedagogical requirements of Section 2.2 to a greater extent.

2.5 Software Which Helps Define Entities and Entity Relationships

Q-MOD (as described in Boohan, 1994), is a calculation program which conveys arithmetic understanding by using linked boxes and arithmetic operators to enable pupils to investigate modelling situations, such as the effect of exercise on body weight. Tools are provided which allow users to create, move, and delete boxes and links, and change the values of variables. Q-MOD visually represents the way in which the variables are related to each other. Therefore, with respect to the pedagogical requirements in Section 2.2, the program helps users identify and define entities from a problem (point 1), and identify relationships between the entities (point 4). NUMERATOR (as described in Boohan, 1994) achieves the pedagogical requirements (points 1 and 4) by using the metaphor of tanks (to represent storage) and pipes (to represent flow through the system) via arithmetic operators. A similar program WORD PROBLEM ANALYST (as described in Kaput, 1992), uses metaphors of "notecards" and arrows. Users input quantities on the notecards, and then the quantities are linked by literally drawing arrows from one notecard to another.

Software, such as: PLANNER (Schwarz, Nathan, and Resnick, 1996); and ANIMATE (Nathan and Resnick, 1993); help users plan and solve word problems. These systems meet two of the pedagogical requirements in Section 2.2. That is, the systems give users control over word problem entities (point 1) and allow relationships between them to be identified (point 4). This is achieved through the use of visual graphics that enable problem situations to be depicted in meaningful ways. An example of a word problem used with PLANNER is: "Five children were playing a game together. Three more children joined them in the middle of the game. How many children participated in the game?" (Schwarz, Nathan, and Resnick, 1996, p. 63). Other software which helps define word problem entities and entity relationships is EDUCE (LeBlanc, and Russell, 1989), TAPS (Derry, Hawkes, and Diefenbach, 1993), and SEMCALC (Schwartz, 1982).

2.5.1 Pedagogical Problems with the Above Software when Representations needed for Calculator Understanding are used to Solve Arithmetic Problems

In a similar way to the author's system described in Chapter 4, Q-MOD, NUMERATOR, and WORD PROBLEM ANALYST, are extremely useful systems insofar as they enable problem situations to be constructed and represented in concrete ways at the interface. However, with respect to the pedagogical requirements of Section 2.2, there are two fundamental distinctions between the above systems and the author's. Firstly, no intermediate representation is used to: (a) facilitate the transition from the concrete to the abstract (point 3 of the pedagogical requirements), (b) identify relationships between entities (point 4), and (c) help transfer from concrete graphics to abstract arithmetic notation (point 5). Secondly, calculation procedures - such as order of operations (point 6) are not addressed where multiple operators form part of an evaluation. This latter problem is highlighted by Kaput (1992), who states with reference to WORD PROBLEM ANALYST, that the "program...does all the computing and inferring..." (p. 536).

2.5.2 Software Which Uses Multiple Representations to Facilitate Understanding

DERIVE (as described in Kaput, 1992) serves as a computational aid for simple and complex expressions, and also makes use of graphing utilities, thereby providing different types of representation to facilitate learning. MATHCAD (Mathsoft Inc., 1995) also uses more than one type of representation, since the system has the ability to compute the values of algebraic statements and then graph them. Such software therefore assists learning by showing abstract symbols in a different and more concretised format. Thus, the pedagogical requirement of converting representations from concrete to abstract (point 5) is met if graphs are considered as being an intermediate representation (i.e. they are neither familiar concrete forms nor abstract notation).

Some software provides a more interactive learning environment. For example, THEORIST (as described in Kaput, 1992) enables algebraic manoeuvres to be carried out via an interface where users are able directly to manipulate algebraic expressions. Thus, variables can be isolated by a user dragging-and-dropping parts of an expression. This system gives users greater control over algebraic manipulations, and could help with the identification of entities in a problem (point 1 of the pedagogical requirements), although the entities would not be represented in a familiar concrete form.

An arithmetic program designed for the early primary age range which utilises concrete visual graphics and direct manipulation, is BLOCKS (Thompson and Thompson, 1990). BLOCKS is a mathematical microworld which uses two notational systems (traditional numeral and Dienes base-ten blocks) as a means of concretising abstract arithmetic notation and subsequent computations. For example, the symbolic number 2462 may be represented in a more concrete way using the expanded blocks: 2 cubes, 4 flats, 6 longs, and 2 units. In order to carry out addition, the computer screen is split into two with a vertical dividing line. Using direct manipulation, various blocks are dragged using a mouse, and dropped in each of the two areas by a user. When *Combine* is clicked, the vertical dividing line is removed, and the blocks are considered as one collection, thus representing addition. This system of visualisation means that the abstract operator symbol (+) is not used, but inferred. The pedagogic requirements (see Section 2.2) of: identifying entities and representing them using a familiar concrete format (point 1); and identifying a class of entities using a concrete format (point 3); are therefore met with BLOCKS. The BLOCKS microworld makes the constraints of decimal numeration explicit because the constraining nature of the two notational systems serves to foster reflection. In other words, users are constrained to compare the actions of symbol notation with the actions using the base-ten blocks, thereby facilitating the building of new knowledge structures. Consequently, this design should enable the concrete blocks to be scaffolded away over time (point 7 of the pedagogical requirements).

Games software, such as WEST (Burton and Brown, 1982), provides both visual graphics and arithmetic notation to support learning. The playing pieces (a stagecoach and a train) race against each other, and through interaction with the game, pupils gain computation skills. A player makes up an arithmetic expression using three computer generated numbers, by inputting addition, subtraction, division, or multiplication operators, as well as parentheses. The user computes the result, and then types the completed statement into the computer (e.g. 3 x (5 + 2) = 21. If the result is correct, the computer moves the playing piece. Pupils gain knowledge regarding computation sequences and order of operations (point 6 of the pedagogical requirements). WEST is an ITS, and during user interaction the system coaches by offering suggestions regarding the computation of arithmetic expressions, so that play may be maximised. Burton and Brown (1982) point out that the computer coach follows a "glass box" approach, because a user's poor move is compared with the requirements needed to make a better move. The coach then offers a better move together with an explanation. Thus, a user can see the association between the explanation and the move. Burton and Brown (1982) point out that WEST is based on a constructivist pedagogy, where the underlying approach is "guided discovery learning" (p. 80).

Other games programs which use concrete graphics and arithmetic notation, but do not employ coaching, include SUBTRACT WITH STICKS and RUBBER STAMP (as described and illustrated in Solomon, 1986). Such programs assist with the identification of entities through the use of concrete icons (point 1 of the pedagogical requirements), and also help with the correspondence between graphics and arithmetic notation.

2.5.3 Pedagogical Problems with the Above Software when Representations needed for Calculator Understanding are used to Solve Arithmetic Problems

A fundamental problem with THEORIST is that it does not provide a situated learning environment to promote meaningful learning, since direct manipulation is simply carried out with conventional abstract algebraic expressions. In addition, with BLOCKS, DERIVE, MATHCAD and THEORIST, entity relationships and arithmetic evaluation procedures with calculators are not addressed once appropriate information has been elicited from the semantics of a problem. Therefore, three fundamental pedagogical requirements of Section 2.2 are not met. Firstly, the relationships between entities (point 4). Secondly, the use of an intermediate representation which lies between the concrete and the abstract formats (point 3) and thirdly, computation procedures (point 6).

The game WEST provides both a situated learning environment and useful feedback regarding computation procedures. However, the following pedagogical requirements are not met if information needs to be extracted from a problem statement prior to computation. Firstly, the identification of entities (point 1), secondly, the use of an intermediate representation which lies between the concrete visual graphics and the abstract arithmetic notation (point 3), thirdly, the relationships between entities (point 4); and fourthly, the conversion of the intermediate representation to the abstract notation used with calculators (point 5). The fundamental problem with software such as, RUBBER STAMP and SUBTRACT WITH STICKS is the lack of an external intermediate representation to facilitate understanding between the use of the concrete visual representations (i.e. dots on stamps and bundles of sticks) and the equivalent abstract symbolic representations (e.g. 5 + 5 + 5 = 15 and 43 - 25, respectively). Thus, points 3 and 4 of the pedagogical requirements in Section 2.2 are not met.

2.5.4 Overall Pedagogical Problems with Current Software for Understanding Calculator Representations

Two fundamental problems remain to be resolved to support the teaching and learning of arithmetic calculator computations based on initial problem statements. Firstly, although current software has gone some way towards meeting the pedagogical requirement for an intermediate representation (e.g. through the use of visually concrete screen images), the fundamental requirement which is not fully met is an *intermediate representation* designed specifically to help with the transfer from word problem information to: (a) an arithmetic expression; (b) arithmetic computations; and (c) the abstract notation of calculators. Secondly, current software does not free users from order constraints with regard to the sequence in which computer procedures (e.g. movement between representations) at the interface have to be carried out.

Having considered the pedagogical requirements and the associated problems with educational software, the following section moves on to outline a solution which meets the pedagogical inadequacies of current software.

2.6 Multiple, Equivalent, Linked Representations (MELRs): an Approach to Satisfying the Pedagogical Requirements

MELRs are particularly suitable for learning and understanding concepts and procedures in the domain of arithmetic. This is because multiple representations present information in more than one form, with the intention that one representation should provide information that the others lack. If different representations are also equivalent and linked, then the computer-based learning environment becomes a powerful educational tool to facilitate pupils' understanding. The terms multiple, equivalent, and linked are described below.

2.6.1 Multiple Representations

Having more than one representation enables information that is lacking in one to be supplied in the other(s) (e.g. a diagram could be used to support text). Thus, multiple systems of representation may be used to facilitate understanding as a result of the actions carried out in one representation being reflected in the other representations. Consequently, when learning arithmetic, if not all aspects of a concept can be adequately represented with a single notation system, other representations may be used to convey information. For example, PLANNER (Schwarz, Nathan, and Resnick, 1996) uses representations that depict concrete graphics and abstract arithmetic notation. The system helps pupils plan and solve arithmetic word problems of the type: "Five children were playing a game together. Three more children joined them in the middle of the game. How many children participated in the game?" (Schwarz, *et al.*, 1996, p. 63). The above problem is represented by a train that is 5 units long (the number 5 is shown above the train). The train is then loaded with 3 units by a load machine (this is added to the end of the train, and + 3 is shown). The final number of children is then represented by a piece of track (on which the train is standing) which is 8 units long (i.e. the train and the added load combined).

2.6.2 Equivalent Representations

In addition to using multiple representations, the conceptual equivalence of different types of representation also helps to promote understanding. Larkin and Simon (1987) distinguish between informational and computational equivalence of representations. Multiple representations are informationally equivalent when all the information in one is inferable from the others, and vice versa. Representations are computationally equivalent if the following two criteria are met: (a) they are informationally equivalent; and (b) inferences drawn from explicit information in one representation can be easily drawn from explicit information in the other representations, and vice versa. Since the information seen in each representation is structurally different (but equivalent), the information will be conveyed differently (but equivalently). This helps pupils make inferences within and between representations, and thus serves to enhance understanding.

An example of the use of equivalent representations is the study by Ainsworth, Wood, and Bibby (1996). They examined the effects of three informationally equivalent multiple representations to assess how the representations influenced the learning of estimation in primary school children aged 10/11 years. The multiple representations were also used to help children learn about their estimation accuracy. The computer-based environment CENTS was used with the following representation categories:

Mathematical (a histogram and numeric display)

Pictorial (a "splat wall" and archery target)

Mixed (archery target and numeric display)

Children in all experimental groups (i.e. mathematical, pictorial, mixed) showed significant improvement at performing estimation. However, children's understanding of their estimation accuracy improved only with the mathematical and pictorial representations. The study concluded that computer-based learning environments should support the user co-ordination of multiple representations in order to promote deeper mathematical understanding.

2.6.3 Linked Representations

Having links between multiple equivalent representations facilitates translation between them and thus serves to further enhance understanding. Linked representations enable an action (either on command or automatically) in one representation to be reflected in another linked representation. Representations which are linked, enable users to transfer between equivalent, but structurally different multiple representations. Kaput (1992) points out that a computer enables a dynamic, interactive medium to operate between linked representations, whereas with other static media, e.g. text and diagrams in books) there are no linkages between actions except cognitive ones. Therefore, with conventional teaching resources, it is not possible to dynamically link representations during learning. Kaput (1989) developed a computer-based learning environment to support multiplicative reasoning in primary school mathematics. The system facilitates concrete to abstract thinking through the use of multiple, linked representations which include familiar icons, text and digits. The concrete icons (e.g. apples) are linked to abstract arithmetic expressions, thus providing a "bridge" to promote understanding. An example of a problem that can be used with the system is: "How many apples will be needed altogether if four children are to get three apples each?" (Kaput, 1989, p. 38).

2.7 Usability Requirements

Usability will be assessed in two ways:

2.7.1 Design Issues Raised from a Cognitive Walkthrough

A cognitive walkthrough is a formalised way of *imagining* users' thoughts and actions when they use an interface for the first time. It is a methodology for performing theory-based evaluations of interface design (Polson, Lewis, Rieman, and Wharton, 1992). Reasons for evaluating a design without users are:

The design should be problem free. In other words, users should not have to waste time on trivial bugs - these should have been caught earlier - hence the cognitive walkthrough.

A good evaluation (i.e. a cognitive walkthrough) can help catch problems that may not be revealed by only a few users.

Shortcomings in the design specification may be uncovered.

Appendix 1 gives further information concerning cognitive walkthroughs. The results of the cognitive walkthrough during the design of ENCAL are given in Chapter 4.

2.7.2 Usability Requirements Using Cognitive Dimensions

The above descriptions indicate that MELRs have the potential for meeting the pedagogical requirements detailed in Section 2.2. To this end, the use of information technology to develop an approach to teaching about number using calculators via the use of MELRs is appropriate. Consequently, in the next chapter a specific MELR approach is developed with the aim of satisfying the pedagogical requirements. However, before this there is a need to address what the usability requirements of the system are.

The usability of the interface is crucial because a well-designed pedagogy will be no use unless it enables pupils to interact in an appropriate way to support learning. Since MELRs are more complex than single representation systems, usability is a much more important issue. The *Cognitive Dimensions* (Green, 1989; Green and Petre, 1996; Green and Blackwell, 1998) approach is particularly appropriate when considering user-system relationships. This is because cognitive dimensions provide a set of terms (i.e., a checklist) that enable the structure of the software to be considered, the pattern of user activity to be described and subsequent improvements to be made. In addition, cognitive dimensions provide a "lightweight" approach to usability, as opposed to more "heavyweight" methods such as GOMS (i.e. Goals, Operators, Methods, and Selection rules). For further information concerning cognitive dimensions refer to Appendix 2.

Green and Petre (1996) discuss cognitive dimensions in the context of visual programming language design when considering the programming operations needed to achieve specific user goals. However, the cognitive dimensions approach applies to all information artefacts (Green, 1989) and is therefore relevant to computer-based learning environments.

The five cognitive dimensions below are particularly relevant to the usability of the software designed by the author in this thesis. They have been considered with regard to (a) user actions carried out at the interface and (b) subsequent system behaviour.

Closeness of Mapping. To support understanding whilst using an interface, mapping between the problem world and the world of the computer-based environment should be as close as possible. According to Green and Petre, the closer the mapping, the easier the problem solving ought to be. The games software SUBTRACT WITH STICKS and RUBBER STAMP provide some closeness of mapping between problem statements and subsequent computer-based problem solving through the use of concrete visual images and abstract arithmetic notation. The TAPS system gives some closeness of mapping, but to a lesser extent since only abstract tree constructions and text are used at the interface. The system in this thesis uses an intermediate representation to facilitate the transition from concrete to abstract representations, and vice versa. The intermediate representation serves to provide a closeness of mapping between problem entities depicted in the concrete representation, and subsequent calculator calculation procedures carried out in the abstract representation. A detailed description of the concrete, intermediate, and abstract representations is given in Chapter 3. The approach outlined in Chapter 3 seeks to maintain a close mapping.

Abstraction Barrier. Abstractions used at an interface may vary in their comprehension difficulty. The abstraction barrier is determined by the least number of new abstractions that must be understood before a system can be used. Notation, text and graphics used at a computer-based learning interface should be easily understandable (i.e. meanings should be more or less self-evident) and require little mental effort to comprehend them. Some of the text output of the Attisha and Yazdani (1984) software is abstract, since unfamiliar terms such as multiplicand and multiplier are used, thus raising the abstraction barrier. In the author's system, the abstraction barrier is reduced because: (a) the interface enables open-ended exploration (i.e. users are not overly constrained); (b) if part of a representation is not well-formed (i.e. an incomplete or incorrect expression has been input), users will be prompted by a graphic to change the existing input; and (c) unfamiliar or complicated text is not used.

Premature Commitment. A user will make a premature commitment at the interface when he/she is forced to make a decision before a required piece of information is available. For example, with the program TOONTALK ARITHMETIC, to add 2 + 3, a user at the interface places the + 3 on top of the 2, and then a mouse runs over to the numbers and smashes them together with a big hammer. The sum of the two numbers is left behind. This system commits learners to enter data in a given order and direction (i.e. towards evaluation) and in a way which may be unfamiliar to their current understanding. Similarly, the use of parentheses with a calculator commits pupils early on to a specific evaluation sequence. The author's system reduces premature commitment by giving users a wider scope of decision-making opportunities regarding evaluations. Thus, users may translate actions at the interface either forwards towards evaluation, or backwards towards the problem statement. In addition, parentheses may or may not be used either early or late in calculation procedures.

Viscosity. Viscosity is resistance to change, and refers to the amount of work a learner has to input at the interface to bring about a small change in the system's behaviour. Thus, the lower the viscosity, the easier it is for pupils to interact and learn. Intelligent software which uses dialogue as the interactive medium (e.g. FITS) tends to have a high viscosity because a list of options or dialogue has to be read, a selection has then to be made or text has to be entered, and then an action is carried out. However, in the author's unintelligent system, viscosity has been reduced at the interface: (a) through the use of multiple, equivalent, linked representations - which means that a single move (e.g. either entering a digit or a graphic) in one representation results in equivalent moves in the other two representations; (b) by allowing users to perform direct manipulation (e.g. a single mouse click will produce a graphic at a required location); and (c) due to user actions being readily reversed.

Hidden Dependencies. A hidden dependency refers to a relationship that exists between two components seen at the computer interface (e.g. text and graphics), where one component is dependent on the other, but the dependency is not fully visible or apparent. For example, MUMATH has hidden dependencies in the computation procedures. Thus, although an answer is dependent on the expression input, the hidden computer algorithm does not explain how the answer is arrived at, and so there is a hidden dependency. Ideally, a system should have no hidden dependencies, since they can hinder comprehension either within a single representation, or between multiple representations. However, a possible drawback of making all dependencies visible is that this may lead to an increase in viscosity (see point 4 above). Within a computer-based learning environment, a trade-off has to be achieved between having too few and too many hidden dependencies if the system is to meet specific educational requirements. In the author's system, hidden dependencies are reduced *between* the three equivalent representations as a result of dynamic linking and concurrent presentation. This enhances usability by facilitating the comparing and subsequent understanding of information between each representation - which needs to be clear from the outset. However, there are some hidden dependencies *within* representations, such as the sequence of computations and the use of parentheses. This has been done in order to foster reflective thinking and/or teacher-pupil collaboration whilst teaching order of operations. That is, these hidden dependencies may be described as follows: a calculation sequence depends upon parentheses if input, but if parentheses are not shown, there is a hidden dependency in terms of what the calculation sequence depends upon.

2.8 Summary

The pedagogical requirements that form the basis of the computer-based learning system are:

- Identify the entities from a problem, and then represent them using a familiar *concrete* format.

- Identify the relationship between one entity from a set, with an entity from a different set.

- Identify a class of entities using the concrete format, and then assign a number to state how many using an intermediate representation which lies between the concrete format, and the abstract symbols used with the calculator.

- Identify the relationships between the entities using the intermediate representation.

- Convert the intermediate representation to the *abstract* notation used with the calculator.

- Carry out computations with the calculator, and use the intermediate representation to assist with order of operations.

- Scaffold the concrete format away over time.

The corresponding usability requirements gleaned from a Cognitive Dimensions analysis are:

- A closeness of mapping is provided between a problem and the computer-based environment.

- The abstraction barrier is reduced at the interface thereby enhancing user-computer interaction.

- Where feasible, premature commitment is prevented with regard to decision making.

- Viscosity is reduced at the interface to facilitate usability.

- Hidden dependencies are omitted between representations, but have been input within representations during computations.

- Usability will also be assessed using a cognitive walkthrough.

In the next chapter, the specific usability of the MELRs is developed with regard to the computer-based learning system.

Chapter 3

The Specific Pedagogical Solution

3.1 Introduction

In Chapter 2 it was pointed out that the general pedagogical solution to the difficulties of understanding abstract calculator representations will best be addressed by providing an interface which uses multiple, equivalent, linked representations (MELRs). In this chapter, the specific solution is addressed with regard to the pedagogical and usability requirements. To this end, Section 3.2 refers to how the author's computer-based learning system - Entities, Notation, Calculator (ENCAL), specifically meets the pedagogical requirements. Section 3.3 outlines two further issues crucial to the learning process. These are user control of the system and the rationale underlying the term computer-based *environment*. Finally, Section 3.4 summarises the chapter.

3.2 Specific Solution to the Pedagogical Requirements

It was stated in Section 2.2 that understanding in individuals is influenced by external situations, and that computers provide a powerful learning medium for using external representations. This is primarily because with a computer, specific representations may be built to suit the educational requirement and thus optimise teaching and learning. This versatility is apparent during problem solving when computer-based representations may be explored to enhance learning using three methods. Firstly, visualisation, since computer graphics enhance our ability to think visually. Secondly, metaphors to promote abstract understanding and thirdly, direct manipulation to simulate the real world through the movement of concrete graphics. Thus, the representations used at the interface can facilitate the linking of visually concrete objects to abstract arithmetic expressions and the construction and use of mental models to promote understanding. Computer-based multiple, equivalent, linked representations (MELRs) are particularly useful in this respect for the teaching and learning of arithmetic, as described below.

3.2.1 Using MELRs to Meet the Pedagogical Requirements

Based on what has been stated in the above paragraph, specific MELRs will be used to meet the pedagogical requirements outlined in Section 2.2. The MELRs will facilitate: (a) the translation of problem information to subsequent computations; (b) the conceptual understanding of an arithmetic expression; (c) the procedures required for calculator evaluations; and (d) the mapping of symbolic information to the formation of mental schema particularly where a user constructs a representation. To achieve these, a problem statement or an arithmetic expression will be depicted using the following *three equivalent and linked representational styles*:

- iconic (real - world, concrete) representation;
- dataflow (intermediate) representation (also referred to as the **datatree**);
- calculator (abstract arithmetic) representation.

Readers from other backgrounds (e.g. psychology or computing) may describe the above representations using analogous taxonomies. For example, the representations may be restated in terms of pictures or icons, structure diagrams (e.g. decision trees), and graphs or tables. Whichever taxonomy is used, the appropriate goals should be achievable (according to the research) in order to predict any limitations of the current visual representations.

In addition, a teacher will need to demonstrate and explain the use and significance of each representation.

3.2.2 Problem Solving Using the Three MELRs

Iconic Representation (Meeting Pedagogical Requirements 1, 2, 3, and 7, Section 2.2)

Real-world icons are used to represent entities from problem statements and the corresponding arithmetic expressions in a more concrete and familiar way. This has been done because psychological theory suggests that individuals construct concrete mental models of entities, and then elaborate these models by manipulating and mentally transforming them (Johnson-Laird, 1983). In other words, a problem statement is related to the real world to facilitate understanding in the minds of children (refer to Section 2.2. for an explanation of mental models). For example, moving book icons to shelf icons simulates the manipulation of objects (i.e. books to shelves) in the real world. This direct manipulation of computer graphics serves three fundamental purposes during learning: (a) it acts as a spatial metaphor; (b) it enables problem entities to be concretised in the minds of individuals; and (c) it enhances the understanding of abstract concepts.

The following specific problem statement highlights the use of the iconic representation.

John has a collection of books. There are many bookshelves in his room. John chooses three shelves and puts four red books on each. John is then given two blue books which he later places on another shelf. How many books has he in his room?

The corresponding iconic representation is shown in Figure 3.1 below.

Figure 3.1. Iconic (Concrete) Representation

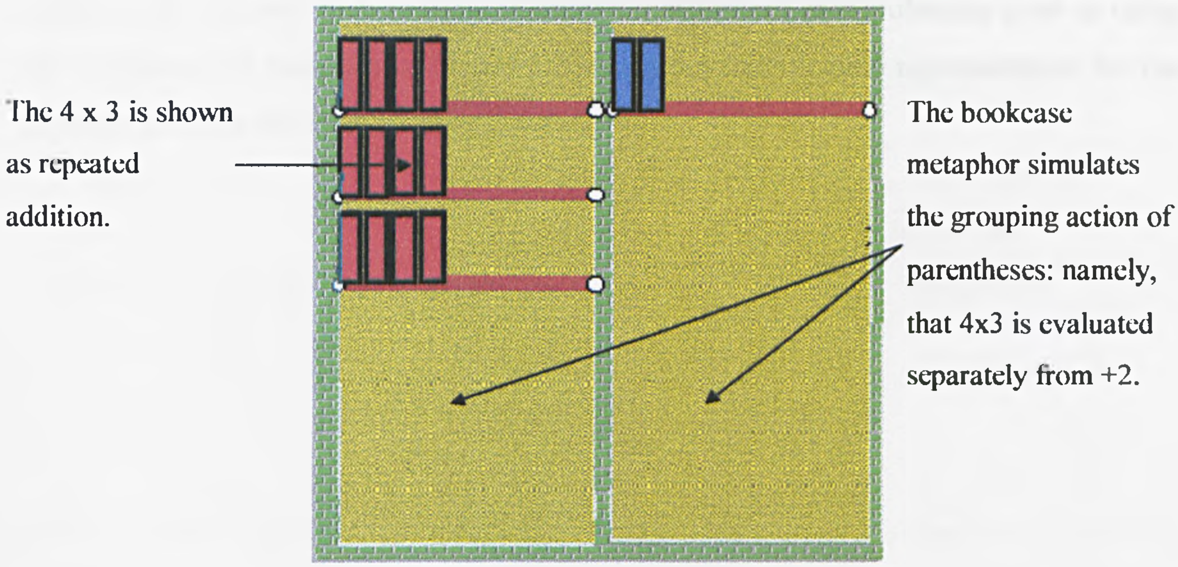
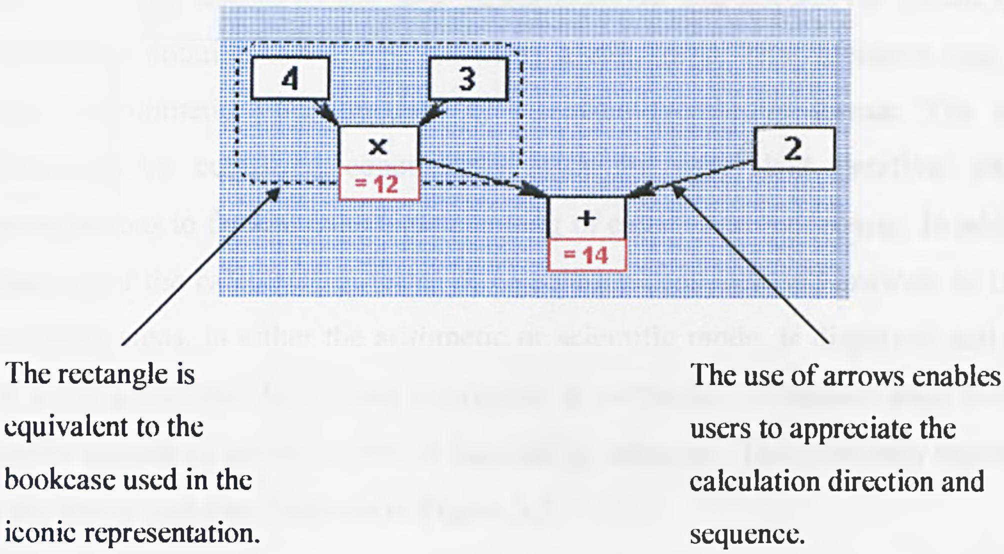


Figure 3.1 indicates how pupils may directly relate the entities described in the problem statement to the computer graphics tools provided at the interface (pedagogical requirement 1). The iconic representation also helps users identify the relationships between sets of entities. In this example, 4x3 (four red books per three shelves) and +2 (two blue books) are shown in separate bookcases indicating separate groupings or sets (pedagogical requirement 2). A user may freely explore with the book and shelf icons to create a concrete picture of the problem. This concrete format may then be used to help construct the intermediate representation (pedagogical requirement 3). Furthermore, the scaffolding provided by the concrete format may be faded over time as users become more proficient with the intermediate and calculator representations (Pedagogical Requirement 7).

Dataflow Representation (Meeting Pedagogical Requirements 3, 4, and 5, Section 2.2)

The dataflow representation facilitates conceptual and procedural understanding between the concrete (iconic) and the abstract (calculator) representations. This is because the dataflow is an intermediate representation that serves as a pedagogic link due to the fact that it is neither wholly concrete nor wholly abstract. The intermediate dataflow representation is designed to help a user more easily translate information both towards evaluation and, if necessary, from evaluation back to the problem. The datatree does this by providing a conceptual bridge (or pedagogical link) between the concrete icons and the abstract symbols that will be used during calculator data-entry. In addition, the datatree enables users to carry out arithmetic manipulations prior to using the calculator for evaluations. Figure 3.2 shows the intermediate representation for the problem given in the iconic representation.

Figure 3.2. Dataflow (Intermediate) Representation



It can be seen from Figure 3.2 that numbers have now been assigned to replace the concrete books and shelves (pedagogical requirement 3). The relationship between entities is also shown using directional arrows (pedagogical requirement 4). In addition, a rectangle is now used in place of the bookcases. Rectangles are more abstract in their meaning than the grouping metaphor depicted by the bookcases in the iconic representation. Specifically, the rectangle shown around the 4x3 in Figure 3.2 indicates two concepts: (a) that this is a calculation step; and (b) this step takes priority in the evaluation sequence. Even if the rectangle metaphor is not used, the 4x3 is grouped on one side of the datatree to help clarify that the multiplication operation needs to be treat separately from the addition operation. This semi-concrete representation may then be used as an aid when the dataflow information needs to be converted to abstract arithmetic notation for calculator data entry (pedagogical requirement 5). It will be noticed from Figure 3.2 that answers are shown for each calculation step (i.e. 12 and 14)[1].

---

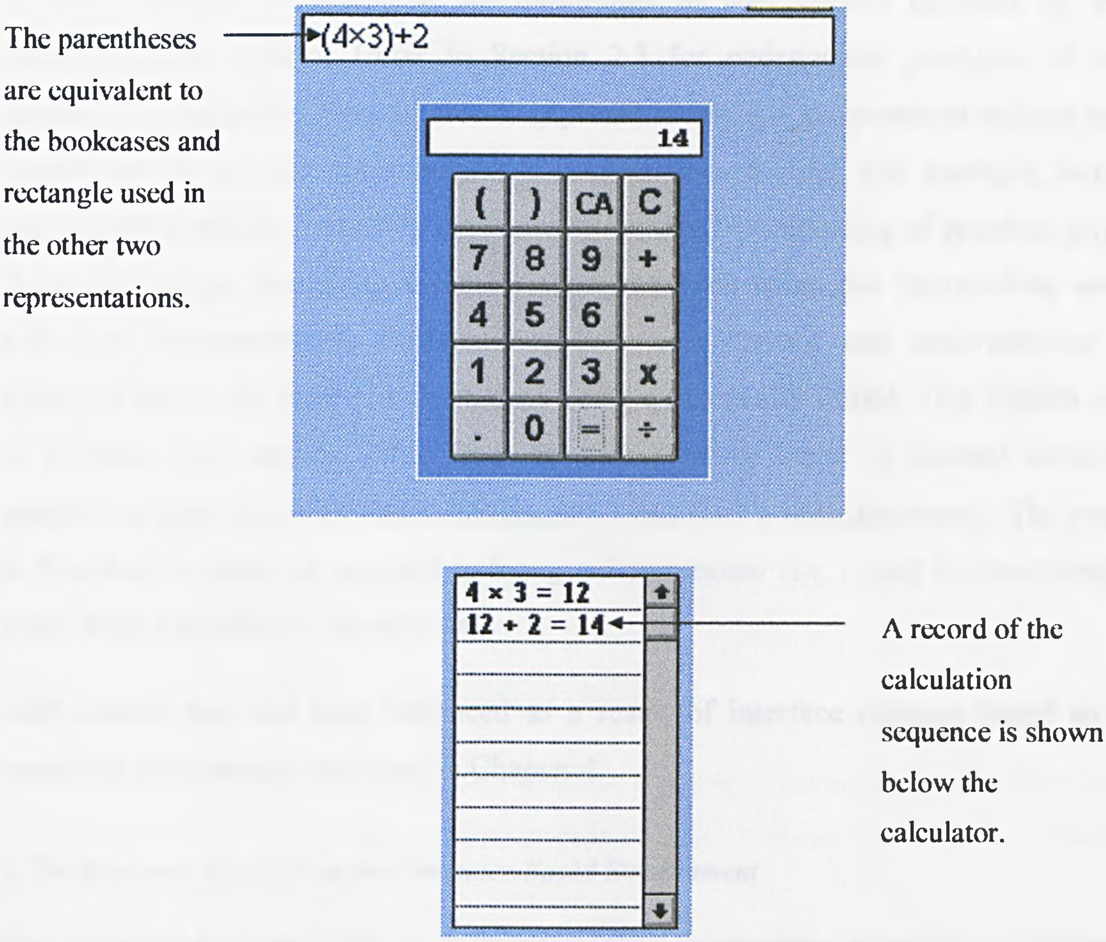
[1] Figure 3.2 is an early version of the system, and it was later decided not to show answers following each calculation step. The reason for this was that users could take computer generated answers literally despite them being incorrect with regard to a question statement.

Calculator Representation (Meeting Pedagogical Requirement 6, Section 2.2)

This is the most abstract of the three representations, and enables the formal system of mathematics notation to be depicted using a calculator. The calculator may be used either in arithmetic (four-function) or scientific (algebraic) format. The calculator syntax can be compared concurrently with the equivalent dataflow and iconic representations to facilitate the understanding of calculation procedures. In addition, the behaviour of the calculator in terms of calculation sequence and answers to individual calculation steps, in either the arithmetic or scientific mode, is displayed and recorded next to the calculator. Users may then return to particular calculation steps to scrutinise possible sources of confusion with a calculation sequence. The calculator representation for the above problem is shown in Figure 3.3.

The abstract arithmetic notation is shown above the calculator in Figure 3.3. The calculator is in the scientific mode, and it can be seen that the complete expression is broken into two calculation steps for the evaluation: (i) 4x3 and (ii) 12+2. This record of events is designed to help users appreciate the logic system used by the calculator in either the scientific or the arithmetic mode. The use of the dataflow representation in conjunction with the calculator is designed to assist with the computation sequence when parentheses are used and the order of operations when no parentheses are used (Pedagogical Requirement 6).

Figure 3.3. Calculator (Abstract) Representation



Pupils may count the book icons in the iconic representation to arrive at an answer, and may not resort to the use of the intermediate or calculator representations. However, the system is designed to be an exploratory, open-ended teaching and learning environment which provides the opportunity to discover how abstract arithmetic expressions are: (a) arrived at from problem statements; and (b) evaluated with a calculator.

3.3. Two Issues Crucial to the Learning Process

Although MELRs are used to meet the pedagogical requirements, two further issues have also been taken into account in order to enhance learning. These are concerned with the amount of control a user has when interacting with the system, and the influence on learning imposed by the computer-based environment. These two issues are described below.

1. User Control of the Computer-Based Learning System

ENCAL attempts to overcome the constraints of user control imposed by other computer-based systems (refer to Section 2.3 for pedagogical problems of other systems). In particular, users are free to explore the learning environment without being constrained to carrying out actions in a pre-determined order. For example, learners may consider entities first using the concrete icons at the beginning of problem solving if desired, or go directly to the calculation sequence using the intermediate and/or calculator representations. Furthermore, both user control and understanding are enhanced due to the three representations being dynamically linked. This enables users to translate their actions either from familiar concrete icons to abstract arithmetic symbols or from abstract symbols to concrete icons (i.e. in both directions). The system is thus *bi-directional* as opposed to being uni-directional (i.e. a user is constrained to move from a problem to the solution).

User control has also been enhanced as a result of interface changes based on the cognitive walkthrough described in Chapter 4.

2. The Rationale Underlying the Computer-Based Environment

The philosophy behind ENCAL is based on the *constructivist* perspective. Therefore, the computer environment only becomes *meaningful* through the process of interaction between a human cognitive system (i.e. user) and the environment. Thus, during interaction users are able to perceive representations/constructs of the environment since these will receive individual meaning by users. Consequently, the environment will influence the learning process.

The learning process is *situated.* Thus, a learner's experience is contained (i.e. situated) within the computer-based environment. Such situated learning provides a more optimal environment for the construction of schema. In other words, what is *being learned* (e.g. an evaluation sequence) is supported by the *learning context* (i.e. the arithmetic world of the computer-based learning environment).

Within the learning environment, *entities* are used as cognitive objects (Greeno, 1983) to facilitate understanding. In addition, the environment facilitates *reflective abstraction* (e.g. arguments during pupil-teacher interaction) which helps the construction of abstract mental structures, *mapping* between representations, and *generalisation* of procedures during problem solving.

External *multiple representations* are used within the learning environment and the movement between them stimulates reflective abstraction (Lehtinen and Repo, 1996). Information in the external representations is *linked* and this helps the development of conceptual knowledge (Hiebert and Lefevre, 1986).

3.4. Summary

Chapter 2 states several pedagogical requirements for understanding abstract symbolic information (i.e. calculator usage) during problem solving. The best solution to meeting these requirements is a computer-based pedagogy specifically involving the use of multiple equivalent, linked representations (**MELR**s).

Two further issues that are fundamental to the learning process are user control of the computer-based environment; and the specific learning environment (i.e. the interface) with which a user interacts.

Chapter 4 considers the design specifications of ENCAL.

Chapter 4

Design Specifications of ENCAL and Design Issues

4.1 Introduction

The design of the computer-based learning environment ENCAL is based on the theoretical assumption that people learn by constructing and reasoning through the use of mental models, whereby understanding and reasoning ultimately become internalised (i.e. implicit). Arithmetic understanding is therefore seen as being a form of cognitive expertise (Greeno, 1991a) which develops through interaction with the computer-based environment. Consequently, the *underlying design aim* of ENCAL is to maximise the quality of learning (Somekh, 1996) as a result of individuals constructing mental schema which will influence future thinking. In addition, the system is designed to promote learning within a social (as opposed to an individual) context by facilitating:

meaningful user-interface interactions;

active thinking (i.e. the actions needed to explore arithmetic concepts and procedures);

interface feedback (using tones and text) during problem solving; and

reflection (individually, with the teacher, and with peers).

Design considerations from a conceptual perspective are set out in the following chapter in order to support understanding of the computer-based system design. The conceptual design considerations stated below reflect the computer program architecture, and are based on the rationale of the above design aim. Initially, Chapter 4 outlines design specification 1 of ENCAL. This is associated with the design of the external and internal representations. The internal representation is described and illustrated from a conceptual perspective in order to highlight its operation despite the fact that the programming used to create this part of the system does not exactly match the idea depicted. The use of iconsworld language is then explained together with conceptual design specification 2. This is concerned with users' movement of icons at the interface. Again, this design is conceptual since it provides an outline of the thinking behind the design for the movement of icons, even though the actual programming procedure does not exactly match the underlying design as shown. An example of iconsworld language is given. The design issues which were considered during the development of ENCAL are then stated including the recommendations following a cognitive walkthrough. Finally, the chapter is summarised.

4.2 Design Specification 1: External and Internal Representations

4.2.1 The External Representations

There are three external representational styles (iconic, dataflow, and calculator) each of which uses its own code taken from the Multimedia ToolBook programming language. Consequently, an arithmetic expression resulting from a word problem may be expressed in three equivalent ways: concrete (icons), intermediate (dataflow), and abstract (calculator). The three representations enable information to be presented in static equivalent and dynamic equivalent ways. Static equivalence means that a change in one representation (e.g. the addition of two book icons) will be reflected in the other two representations, such that the end-state of all three is equivalent. Dynamic equivalence implies that any action (i.e. the static state is altered) which occurs in one representation will be reflected in the other two representations, such that there is a correspondence between actions in the three representations. For example, dragging and dropping two books onto one shelf in the iconic representation will result in corresponding actions in the dataflow and the expression above the calculator. Dynamic changes may be unobservable in all three representations concurrently at the interface, however such changes do occur as may be seen in the resulting static states. Static and dynamic equivalence form part of the underlying architecture of the ENCAL computer-based learning system.

4.2.2 The use of Multiple Equivalent Linked Representations (MELRs)

The interface makes use of MELRs (multiple equivalent external computer-based representations) which are linked[2]. From an educational point of view, the use of MELRs will enable users to transfer between equivalent, but structurally different representations, and since the information seen in each representation will be structurally different (but equivalent), the data will be conveyed differently (but equivalently), and this will serve to enhance understanding.

Concurrent Views and Order of use

A user will be able to have concurrent views of all three external representations. This will facilitate understanding concerning the translation of information from problem statements to evaluation, the mappings between representations and the sequence required for data entry into the calculator. In addition, the representations may be used in any order. The order used will depend on the way a user thinks, and also on his/her ability. For example, some users may have no need to use the iconic representation, whereas others may use it to acquire a more explicit understanding.

Design of the Concrete Iconic Representation

The benefit of using icons from an educational perspective is that icons represent objects and their actions that can be manipulated by a user. The design of the iconic representation uses a strong and familiar *metaphor* of books, shelves and bookcases as a means of providing users with the opportunity to represent problem *entities* (e.g. selecting three red books, eight blue books, and four shelves) concretely before considering the *relationships* between them. Thus, the way a pupil approaches a problem is not constrained by abstract operators which connect numbers as is the case with an algebraic expression.

However, a design and educational drawback with using such a concrete representation is that it is not recursive. That is to say, bookcases cannot contain other bookcases. The

---

[2] Refer to Chapter 2 for an explanation of multiple equivalent linked representations.

icons are therefore unable to represent such expressions as 2x3x5 in a literal format. This is because 2x3 is represented as two books on each of three shelves (repeated addition) in one bookcase. The x5 cannot be shown within the same or an adjoining bookcase since only five shelf icons are stated. Consequently, the use of icons can constrain arithmetic expressiveness due to their concreteness. The advantage of using icons on the other hand is that they enable relationships to be seen between the entities of expressions such as: (2x(3+5)). Currently, the iconic representation enables relationships to be shown where expressions include multiplication and/or addition operators. The iconic metaphor therefore supports operations that are not inverses of each other. This is because most confusion occurs during evaluations when mixed operations are used which are not inverses of each other (e.g. + and x) (Ecker, 1989). Some users may decide to represent entities, but not show relationships between them using the icons. Therefore, a less concrete (i.e. intermediate dataflow) representation has been used to overcome the constraints of icon use. The intermediate representation is described next.

Design of the Intermediate Dataflow Representation

The dataflow representation is a recursive tree structure, and has been designed to provide a conceptual bridge between the concrete icons and the abstract algebra used during calculator calculations. The datatree is more recursive than the concrete icons, and enables relationships between entities to be achieved more readily through the use of connectors. As with the iconic representation, the datatree enables entities to be selected before the relationships are added. However, the advantage of the datatree is that there is less constraint on the relationships which may be achieved between entities due to the fact that the trees are less concrete.

A "box" *metaphor* is used to represent the hierarchy in which operations are carried out. The reason being that such comparisons help pupils to learn (Carroll and Mack, 1985). The box or boxes (i.e. transparent rectangles) positioned over parts of an expression indicate the order in which an expression should be evaluated. Thus, users should enter data into the calculator representation according to the rules of arithmetic, and not simply from left to right. Although the box metaphor does not provide any more information than parentheses themselves, it enables transparent boxes to be used initially to "dress-up" an expression to get across its meaning. Initially, transparent boxes were to be used for immediate attention, and the use of opaque boxes was considered for parts of an expression that required delayed attention. However, the use of opaque boxes was ruled out because users would be unable to see a whole expression, and this could inhibit learners' interpretations and understanding during arithmetic evaluations.

With the ENCAL design, the third and most abstract way of representing a problem statement is through the use of algebra and a calculator. The calculator representation is considered next.

Design of the Calculator (Abstract) Representation

With an algebraic (in this case arithmetic) expression, the entities and relationships between the entities cannot be represented separately because of the constraining format of the written expression. This format provides little information regarding the relationships between entities, the translation of information from problem statement to arithmetic expression, and the concepts and procedures required to evaluate an expression. Algebraic expressions are shown above the calculator, as may be seen in Chapter 3. In order to assist understanding, prior to the entering of algebraic data into the calculator representation, users will have represented a problem using the icons and/or datatree so that the entities have been selected, and relationships between the entities have been established. Consequently, the sequence with which abstract algebra is to be entered into the calculator, and the relationship between numbers and operators, should be better understood. Whilst concurrently viewing the dataflow and calculator representations, users can thus input data into the calculator based on arithmetic order of operation procedures.

Two types of calculator logic (i.e. four-function (arithmetic) and algebraic notation (scientific) have been programmed for use at the interface depending on the educational requirement. A switch is provided at the interface to allow the user or experimenter to choose.

4.2.3 Conceptualisation of the Internal Core Representation

ENCAL has been programmed using Multimedia ToolBook. From a *conceptual* point of view, the system works such that an internal core representation serves as the general manager of the system in that it interprets data from, and relays data back, to the three external representations (i.e. iconic, dataflow, and calculator). The internal core representation ensures that the information which comes from the external representations is translated into a language they can all read. The primary aim of the internal core representation is to ensure that static and dynamic equivalence is maintained in the three external representations. The relationship between the internal core and the external representations is shown in Figure 4.1.

Figure 4.1. Relationship between the Internal and the External Representations

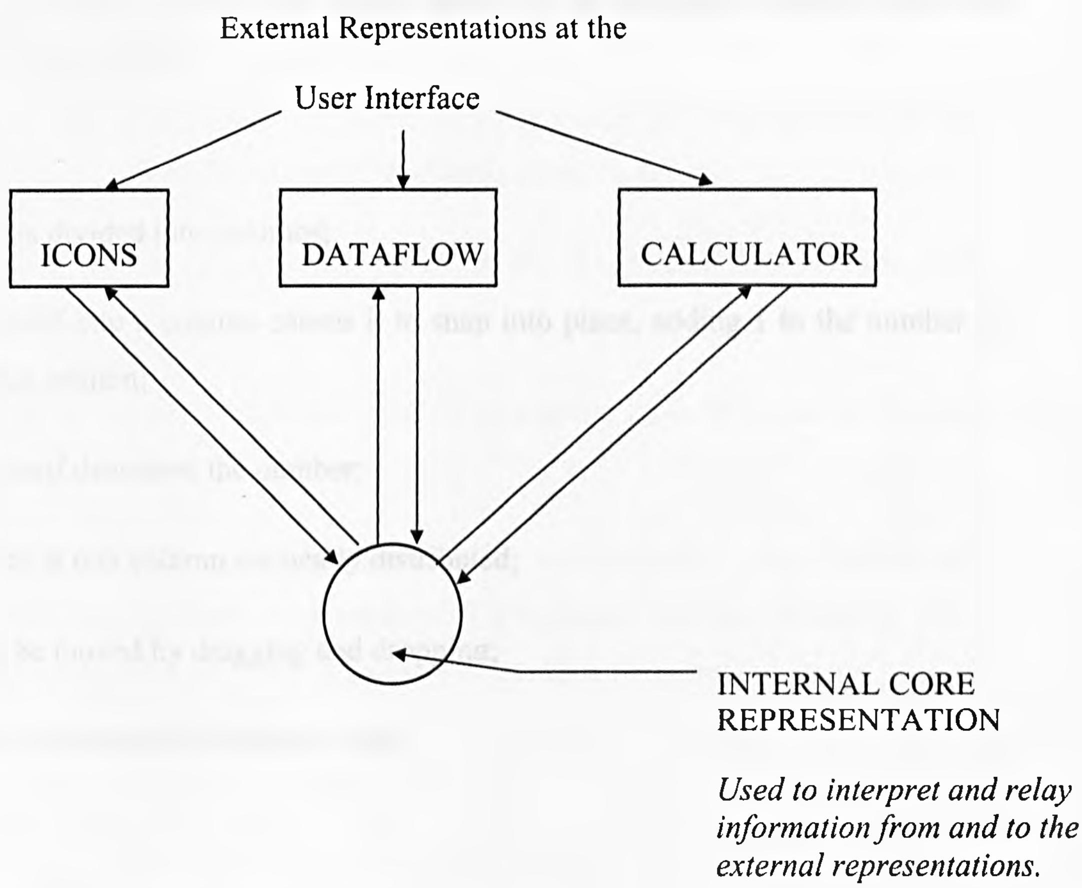


Figure 4.1 indicates that users can read and manipulate any of the three external representations (i.e. iconic, dataflow, calculator) at the interface, whilst at the same time the internal representation ensures that concurrent static and dynamic equivalence is maintained between the external representations during user input.

4.3 Design Specification 2. The use of Iconsworld Language

To enable information in the concrete iconic representation to be translated equivalently to the other two representations, a language (iconsworld) was developed for describing the book and shelf icons.

4.3.1 Conceptualisation for User Actions in Iconsworld

The following design specification was provided in order to devise the iconsworld language. The specification below provides a conceptual framework of how the system operates, despite the fact that the actual design of the computer program does not exactly reflect this outline.

SHELVES:

the interface is divided into columns;

releasing a shelf into a column causes it to snap into place, adding 1 to the number of shelves in that column;

removing a shelf decreases the number;

all the shelves in one column are neatly distributed;

shelves may be moved by dragging and dropping;

shelves may be deleted with a button click.

BOOKS:

books cannot be in a column with no shelf;

releasing a book onto an empty shelf in a column makes it sit at the left side – successive books go side by side in that column;

in a column with a missing shelf, a book snaps to the next shelf down and goes next to the previous book;

when a shelf is full, a message is displayed or a warning tone is emitted;

books may be moved by dragging and dropping;

books cannot be deleted, but may be removed from a shelf and placed in an area outside the bookcases.

WELL-FORMED COLUMNS

During user interaction in the iconic representation, the books and shelves placed into bookcases must be well-formed if the system is to read the input and translate the information to the dataflow and calculator representations. It was therefore decided a column is well-formed only if:

- it has one shelf, and all books are either the same colour (i.e. blue or green) or some books on the shelf may be blue and some books may be green;
- if more than one shelf is used, the number of books of each colour on each shelf is the same.

Example: Figure 4.2 shows books and shelves as may be input by a user in the iconic representation.

If unequal numbers of books are placed on shelves in a bookcase, then remedial action will need to be taken by a user. In some cases it is expected that teacher support will be required in order to assist with this re-forming of books and shelves.

4.3.2 Iconsworld Language for Describing Icons

The language that describes the icons in Figure 4.2 is stated below. The programming utilised does not conform exactly to the procedure outlined below, however this is a conceptualisation of how the system has been designed to operate.

Figure 4.2. Books and Shelves Language in the Iconic Representation

[Number of columns:] **2**

[For column 1:]

[Number of shelves:] **1**

[For Shelf 1:]

[Number of books:] **2**

[Colour for book 1:] **green**

[Colour for book 2:] **green**

[For column 2:]

[Number of shelves:] **3**

[For shelf 1:]

[Number of books:] **3**

[Colour for book 1:] **blue**

[Colour for book 2:] **blue**

[Colour for book 3:] **green**

[For shelf 2:]

[Number of books:] **3**

[Colour for book 1:] **blue**

[Colour for book 2:] **blue**

[Colour for book 3:] **green**

[For shelf 3:]

[Number of books:] **3**

[Colour for book 1:] **blue**

[Colour for book 2:] **blue**

[Colour for book 3:] **green**

The iconsworld language therefore reads as follows:

2, 1, 2, green, green, 3, 3, blue, blue, green, 3, blue, blue, green, 3, blue, blue, green.

4.4 Design Issues

The following design issues were considered during the development of the ENCAL system.

1. Which Arithmetic Operators should be used?

The choice of arithmetic operators is an issue that arose during the design of the system. So far, only addition and multiplication operators have been considered for use in the design. This is because:

Most confusion occurs during evaluations when mixed operations are used which are not inverses of each other (e.g. + and x) (as mentioned above - Ecker, 1989).

The + and x operators can be represented using the book and shelf icons. Subtraction and division would be much more difficult to represent in a literal format using the icons. A possible solution is that a different icon world could be used to represent each of the four operators separately. Thus, relationships could be shown between entities which involve one or more operators which are the same (i.e. either all ÷, or all x, or all +, or all -).

Although the restriction of using only addition and multiplication is significant, the aims of the study can nevertheless be evaluated using these operators. This issue will be considered further in Chapter 10.

2. How Many Representations will be Shown at One Time?

OPTIONS:

only one;

any two;

controlled two (e.g. always dataflow and one other);

all three.

CRITERIA CONSIDERED:

It was thought that:

Controlled two would encourage the use of dataflow as a translation medium.

All three may lead to cognitive overload (i.e. too much choice may obscure (Plowman, 1992) and interfere with the learning experience since the underlying structure may be inappropriate for a person's existing schema, and thus the adoption of new schema and learning will not take place).

Ease of use needs to be a fundamental consideration.

It was decided that all three representations would be user active and shown concurrently because it was felt this would be of greater educational value during problem solving. However, bearing in mind the need for a balance between information presented and choice, users have been given the option of working with only one representation at a time. In this case, all three representations are shown, but hatching is placed over the two not in use. Concurrent information is still displayed, but the two representations not used are user inactive.

3. The use of the Book, Shelf, and Bookcase Icons

A choice has been made as to whether 3x4 is represented by 3 shelves, 4 books, or vice versa. It was decided to represent the number that precedes a multiplication operator by using shelf icons. The number after a multiplication operator is represented by the book icons. Thus, with the expression: 1x5, the 1 is represented by one shelf, and the five books represent the 5. The book and shelf icons have been used in this way because if 2x5 is subsequently input, the number of books would remain the same, but the number of shelves would change to two. This avoids the confusion of the five books on one shelf changing to two books on five shelves. Further comments about this arrangement are given in Chapter 5.

The bookcase icons represent and simulate the grouping function of parentheses used in arithmetic expressions. The action of the system will depend on which calculator has been selected (i.e. arithmetic or scientific). For example, in the scientific mode, the expression 2x(3+4) if input directly would be represented by two shelves each with seven books split into two sets (e.g. four blue books and three green books), in one bookcase. The expression 2x(3+4) would thus be grouped within one bookcase to signify that this bookcase is evaluated separately from any books and shelves in other bookcases. The bookcases do not indicate the priority for evaluating parts of an expression, because at this stage there is no need since users can count the books in bookcases to determine an answer. Similarly, if 2x3+4 is input while in the scientific mode, the expression is represented by two books on each of three shelves in one bookcase, and four books on one shelf in the next bookcase. The fact that two bookcases are used indicates that 2x3 is grouped and thus evaluated separately from +4, thus following the order of operations rule. A user may count the total number of books in the two bookcases that would reveal the answer of ten. As a further example, if 2+4x3 or (2+4)x3 is entered while in the arithmetic mode, the system reads the expression from left to right. Consequently, 2 green books and 4 blue books are grouped on each of three shelves in one bookcase giving the answer of 18. However, 2+(4x3) would result in 2 books appearing in one bookcase, and 4 books on each of 3 shelves appearing in another bookcase, giving the answer of 14.

4. Two Possible Datatree Designs

DESIGN 1. When a datatree is constructed, the rectangle metaphors that simulate parentheses could be input automatically by the computer. With this design, a user would not be required to "dress up" an expression (i.e. decide which part to evaluate first).

DESIGN 2. A variation of design 1 would enable greater user interaction. That is, users could select one or more parts of an expression and input rectangles to represent parentheses.

WHY DESIGN 2 IS PREFERRED. Design 2 is considered to be better than design 1 because it will help users to: (a) think before carrying out actions; (b) generate questions by themselves, to teachers, or to peers; and (c) facilitate the implicit learning of concepts and procedures associated with order of operations.

5. Should Answers be shown After Each Calculation Step in the Datatree?

It was decided not to show answers after each calculation step in the datatree because:

Answers would appear automatically even before data had been entered into the calculator and the equals key pressed.

Equivalence between the three representations would not be maintained since the datatree only would show answers.

Showing answers early could be potentially confusing and perhaps wrongly influence user thinking.

It was thought that pupils could assume that equals as shown would signify "makes" something.

6. The use of one or two Separate Activity Areas

A fundamental design decision was whether a shelf should be input by the system with expressions of the type: 3+2. If not, then books would stand at the bottom of bookcases instead of on a shelf or shelves. Consequently, with the expression: 3+2, three books would sit at the bottom of one bookcase, and two books would be in another. Therefore, from a user's point of view, the bottoms of bookcases would behave like shelves and this could be potentially confusing. This raised another question. Should books that are not on shelves be placed in a separate area outside the bookcases (e.g. as with 3+2), or should all books be placed on shelves in the bookcases area, so 3+2 becomes 1x(3+2)? It was decided that all books would appear on shelves when in bookcases. However, it was also decided to give users the freedom to play around with and place "loose" books outside the bookcases on a carpet area if desired, but these books would not affect calculations until placed in bookcases. Thus, only books and shelves within bookcases would affect calculation outcomes.

A possible drawback with having two separate areas is that pupils may wonder why some books are placed outside bookcases and others are placed inside bookcases. However, the benefit of this design is that pupils can select and manipulate books without shelves prior to placing books in a bookcase. A benefit of having a carpet area is that it provides a space where users can place books, thus avoiding the confusion of books being placed at the bottom of bookcases. Another benefit of the carpet area is that it gives users the freedom to interact with problem entities in any order - that is, books before shelves, or shelves before books. The fundamental benefit of this decision being that the constraints of user interaction are reduced. The advantages of placing all books on shelves in the bookcase area are as follows:

- to assist the understanding and evaluation of calculations;
- the confusion of having separate activity areas (i.e. carpet and bookcases) for calculation purposes is avoided;

- the bookcases, books, and shelves can be used to represent the actions of parentheses or order of operations, whereas the use of the carpet area would be less effective and more confusing;

- the layout of arithmetic expressions within the single bookcase area provides greater consistency for conceptual understanding.

7. *Translation of Information from Icons to the Datatree*

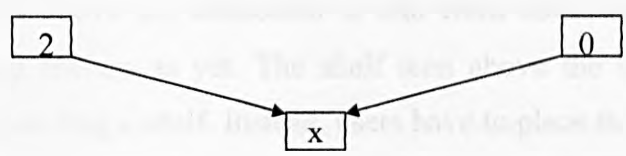
It was thought that users would have difficulty translating information from the iconic representation to the dataflow representation. Consequently, the following solution was considered for the dataflow representation. Firstly, the number of books could be shown over a book icon, and the number of shelves could be displayed on a shelf icon. Secondly, bookcases could be shown behind the dataflow calculation steps of an expression. Thus, for the expression: 2+3x4, a bookcase would be shown behind the 2, and another bookcase would be placed behind the 3x4. The + operator would be displayed between the two bookcases. Thirdly, there would be a large bookcase behind the two bookcases and the + operator. It was thought this could help signify addition. This is because the user would see that the two smaller bookcases need to be combined (by adding) to make the larger bookcase. Fourthly, multiplication could be signified by duplicating the contents of a bookcase. For example, with (2+3)x4, the x represents copy, and the bookcase containing the 2+3 would be duplicated four times.

Despite the possibility that translation between the iconic and dataflow representations would be more easily understood, it was decided not to implement the above changes to the dataflow representation. The fundamental reason being that the different representations should not overlap, but instead signify independent aspects of arithmetic. This would allow each representation to be understood in its own right and so avoid being confused with parts of another representation. However, it was thought that the operator and the number "boxes" in the dataflow representation should be a different shape and/or colour to signify that there is a difference between their actions. Thus, it was decided that the operators should be enclosed in red-filled rectangles, and the numbers would remain in white-filled rectangles. As a result of changing the dataflow symbols, the operator buttons on the calculator were coloured red to correspond to the red rectangles used in the dataflow. In addition, green (as opposed to red) books were used.

8. Several design issues raised following a cognitive walkthrough of ENCAL

(a) AVOIDING THE USE OF ZERO IN THE DATATREE AND THE COLOUR-CODING OF SHELVES AND BOOKS. It was suggested that the books and shelves could be colour-coded in the dataflow representation. This would then avoid the need to show zero in the datatree. For example, 2x0 in the iconic representation is depicted by two shelves with no books on them in one bookcase. The 2 represents the shelves, and the 0 represents the books. 2x0 in datatree format is represented in Figure 4.3 below.

Figure 4.3. 2x0 in Datatree Format



In addition, if only one bookshelf is placed in a bookcase, currently a zero is shown in the datatree. In other words, no value is depicted for one shelf in a bookcase. The reason for this is because of programming difficulties encountered using ToolBook software. However, the ideal situation is described below following the cognitive walkthrough.

Two modifications were suggested. Firstly, the shelf "box" showing 2 could be coloured black with the 2 shown inside, as above. Secondly, the book "box" could be coloured green or blue depending on which coloured books had been selected, but would be empty (i.e. no zero would be shown) since no books were selected. It would then be more obvious and meaningful that there are no books if the appropriate coloured book "box" was empty. This would correspond more closely to the iconic representation that would show no books, but two empty shelves in this example.

A further problem that arises in the datatree representation when there are zero books, is the background colour of the boxes. Ideally, they should not be either green or blue since these colours are used to represent sets (i.e. numbers) of books. Therefore, in order to provide a visual distinction between numbers of books and no books, a unique colour for a box with zero books needs to be adopted, such as yellow and black stripes.

(b) UNDO. A button is needed which when clicked will undo or restore previous actions in all three representations.

(c) TICKS AND CROSSES. The use of ticks and crosses to represent well-formed in each representation should be changed to different graphics (i.e. more user-friendly, such as happy/sad faces). In addition, users may confuse the ticks and crosses with work marked as being correct or incorrect.

(d) DRAGGING, DROPPING, AND BUTTON CLICKING. Users drag and drop books from the icons above the bookcases to and from bookcases. However, users cannot drag and drop shelves as yet. The shelf icon above the bookcases gives the impression that you can drag a shelf. Instead, users have to place the mouse pointer in a bookcase and do a left button click of the mouse for a shelf to appear.

(e) HIGHLIGHT BOOKCASES. When the mouse pointer is positioned in a bookcase, the bookcase could be highlighted to signify the object in use - currently this does not happen.

Issues (c) and (d) were implemented prior to the pilot test because these clearly needed alteration in order to avoid confusion at the interface. The remaining design issues were considered, and in some cases implemented, only after the pilot test had been carried out. This gave the opportunity for the existing design to be user tested thus enabling more informed decisions to be made based on the results.

The rationale underlying cognitive walkthroughs in the design process is described in Appendix 1.

4.5 Summary

The design specification of ENCAL has two aspects to its underlying architecture. The first is associated with the three external representations: iconic (concrete), dataflow (intermediate), and calculator (abstract). These three representations: (a) are linked; (b) enable information in each to be depicted equivalently; and (c) are viewed concurrently to facilitate understanding. The second is the internal core representation which ensures static and dynamic equivalence is maintained in the three external representations. To achieve equivalence, iconsworld language was developed to enable Multimedia ToolBook to read and carry out actions when icons are used at the interface. Throughout the design process several design issues have been considered. These have had a major influence on the design outcome, and include the following:

How many representations will be shown at one time?

Which arithmetic operators should be used?

The use of the book, shelf, and bookcase icons.

Two possible datatree designs.

Should answers be shown after each calculation step in the datatree?

The use of one or two separate activity areas.

Translation of information from icons to the datatree.

Design issues raised from a cognitive walkthrough.

Chapter 5 considers the design process and the pilot evaluation of the initial version of ENCAL. Conclusions of the pilot test and subsequent design changes are stated.

Chapter 5

Design Process: Storyboards, Pilot Evaluation, and Redesign

5.1 Introduction

It is assumed that there is a close relationship between interface design and learning as suggested by Schär (1996). The subsequent effects on learning and understanding are therefore seen as being determined as a result of users interacting with a particular design of computer-based environment. The design specifications covered in Chapter 4 will influence user interaction and thus learning. Instead of concentrating on design specifications and issues raised, the following chapter describes the design process in the form of storyboards, testing and redesign. Initially, storyboard version 1 is set out which includes explanations and drawings concerning the interface design (i.e. the three external representations). The ENCAL system was subsequently programmed to reflect this storyboard (see Figure 5.5). The pilot study is then outlined including the results. The pilot study was carried out using the interface that resulted from storyboard version 1. Based on the results of the pilot study, storyboard version 2 and the revised computer-based environment were developed. The explanations and drawings regarding storyboard version 2 are outlined. Finally, a summary is given.

5.2 Storyboard Version 1

5.2.1 Movement Between Representations

To enhance understanding of problems, it is expected that a user's preferred movement between the three representational styles at the interface would be from concrete to abstract as follows:

1. *icons* (concrete shapes) to dataflow;

2. *dataflow* (intermediate structure) to calculator;

3. *calculator* (abstract notation) to dataflow (the dataflow representation depicts what happens inside the calculator during key presses - so this will serve as the calculator *mental model*).

However, progression between the representations need not always proceed from concrete to intermediate to abstract. Instead, movement may for example be concrete to abstract to intermediate. In fact, users may employ any order to suit their understanding.

A user is able to depict a problem statement starting with: (a) icons (coloured rectangles), (b) datatree (connecting arrows, "boxes", numbers, and rectangles), or (c) calculator (algebra). In the iconic and datatree representations, a user constructs a two-dimensional diagram using the tools provided. Whichever of the three representations is used to construct and depict a problem, the remaining two representations will automatically be constructed and shown at the interface. This is because all three representations are equivalent and linked, and therefore changes in one will be reflected in the other two. This helps promote concrete to abstract understanding. Each of the three representations is illustrated below using the following example problem:

There is one shelf in one bookcase in John's room.

There are three books on the shelf.

In another bookcase there are five shelves of books.

On each shelf there are four books.

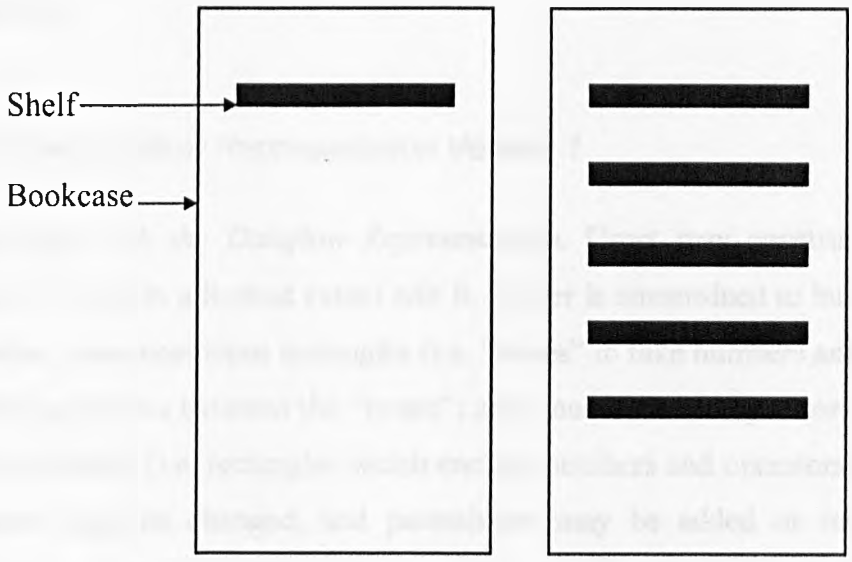
How many books are there in John's room?

5.2.2. The Iconic Representation Version 1

Interacting With the Iconic Representation. Four bookcases have been pre-programmed to appear at the interface when a user enters the learning environment. The book and shelf icon dispensers also appear and are situated above the bookcases (see Figure 5.5). Book and shelf icons may be dragged from the dispensers and dropped in the bookcases. A user places the mouse pointer on the required icon and holds down the left mouse button in order to drag. Book but not shelf icons may also be dragged from bookcase to bookcase. This enables a person to rearrange books on shelves. Shelf icons have to be placed in a bookcase before book icons will be accepted.

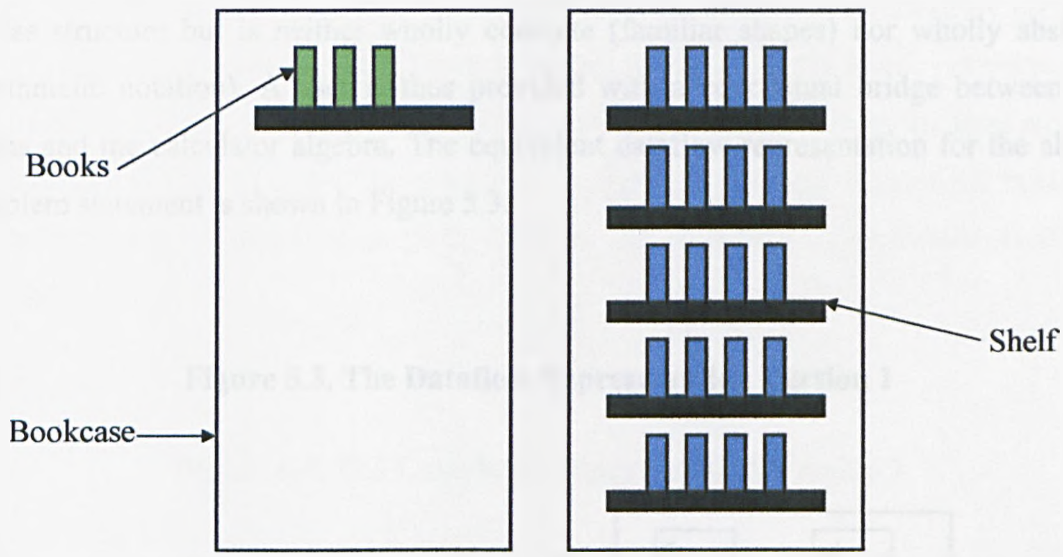
Assume a student decides to represent the above problem using *icons*. The appropriate number of shelf icons first needs to be dragged and dropped into the two bookcases. This is shown in Figure 5.1.

Figure 5.1. The Iconic Representation Version 1 (i)



The book icons may be used to split an expression into *sets* with the use of different colours. To achieve this, the multiplication step first needs to be transformed into *repeated addition*. The above expression would therefore be arranged in the iconic representation as shown in Figure 5.2.

Figure 5.2. The Iconic Representation Version 1 (ii)



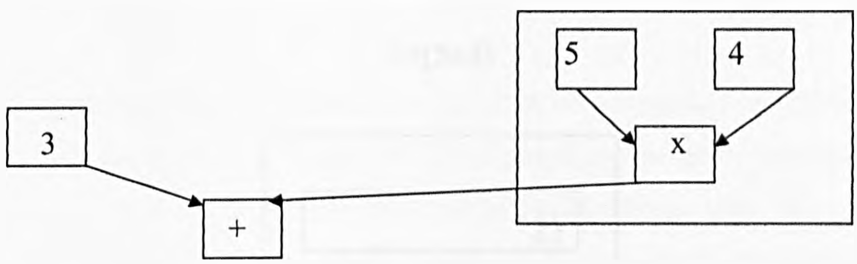
The book icons are dragged and dropped into the bookcases as indicated in Figure 5.2. Using the icons in this way, a concrete representation of the problem statement is established.

5.2.3 The Dataflow Representation Version 1

Interacting With the Dataflow Representation. Users may construct a datatree (see Figure 5.5) and to a limited extent edit it. A user is constrained to building a tree in the following sequence: input rectangles (i.e. "boxes" to take numbers and operators), input connecting arrows between the "boxes"; enter numbers and operators into the "boxes"; add parentheses (i.e. rectangles which enclose numbers and operators). Numbers and/or operators may be changed, and parentheses may be added or removed. However, connecting arrows and "boxes" cannot be moved. Menus at the interface and the left/right mouse buttons enable dataflow construction and editing.

The above problem statement may initially be represented by constructing a datatree as opposed to using icons. However, in order to construct a datatree representation of the above problem, a user will first need to form the appropriate arithmetic expression which in this case is 3+(5x4). As previously stated, the underlying rationale is to facilitate concrete to abstract understanding. Therefore, the preferred approach is to construct the icons as above and then observe the automatic construction of the corresponding datatree. The datatree thus serves as the intermediate representation since it has structure but is neither wholly concrete (familiar shapes) nor wholly abstract (arithmetic notation). A user is thus provided with a conceptual bridge between the icons and the calculator algebra. The equivalent dataflow representation for the above problem statement is shown in Figure 5.3.

Figure 5.3. The Dataflow Representation Version 1



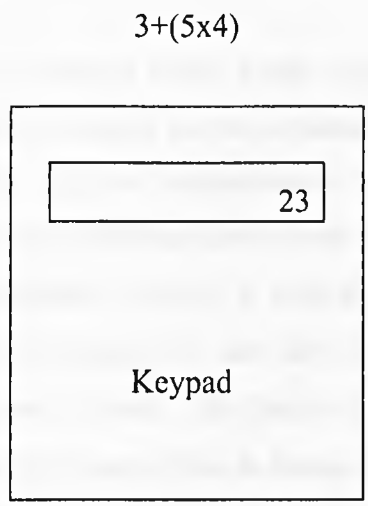
The dataflow representation in Figure 5.3 is shown concurrently with the iconic representation to assist users' interpretation.

5.2.4 *The Calculator Representation Version 1*

Interacting With the Calculator Representation. A calculator keypad automatically appears at the interface (see Figure 5.5) once a user has entered the representational environment. A four-function (arithmetic) or an algebraic notation (scientific) calculator is linked to the iconic and dataflow representations. The equivalent algebraic expression may be shown above the calculator and this is based on the calculator logic and the initial user construction using the icons or datatree. Thus, the arithmetic expression shown above the calculator corresponds to user input. That is, either the calculator keys clicked by a user or the equivalent expressions resulting from the iconic or datatree constructions. If the Look Inside button is clicked, the calculation steps with corresponding answers are shown. An arithmetic expression may be deleted symbol by symbol using the cancel button. New symbols including parentheses may be input using the calculator keypad.

With regard to the problem statement given above, the answer is achieved by pressing the "equals" key on the keypad. A user can count the books in the iconic representation to check the calculator answer. The calculator answer may differ from the total number of book icons if the calculator behaviour (i.e. either four-function or algebraic notation) is inappropriate for evaluating a given problem. The calculator representation is shown in Figure 5.4.

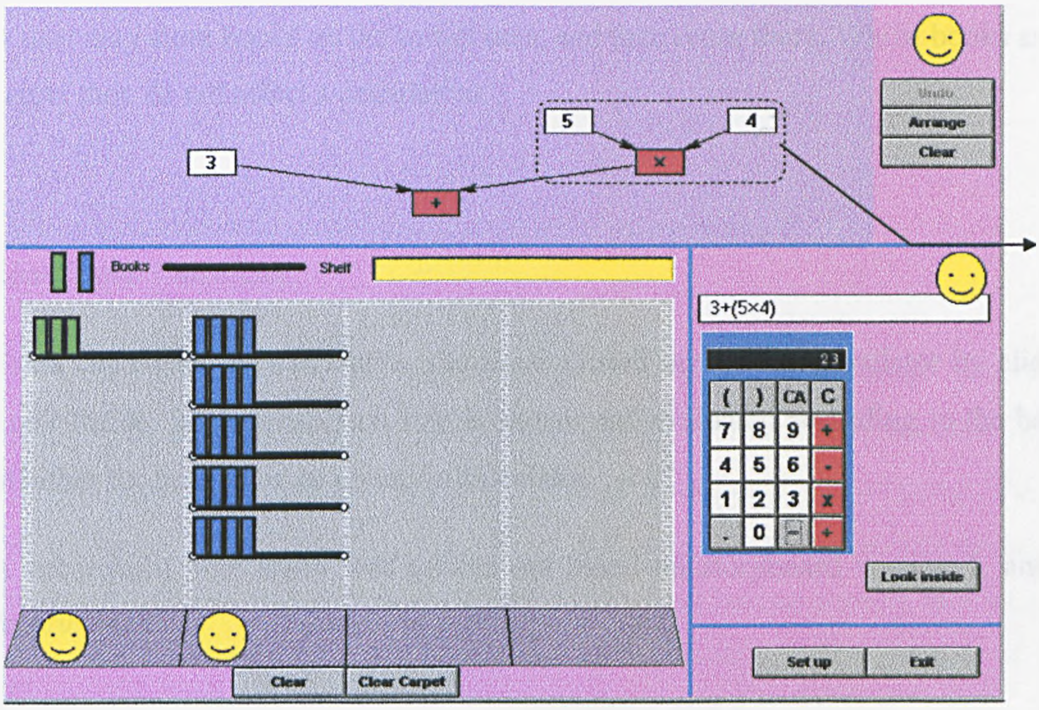
Figure 5.4. The Calculator Representation Version 1



A user may refer to the datatree (see Figure 5.3) to appreciate the calculation sequence of the calculator.

Figure 5.5 shows the three representations of storyboard 1 (iconic, datatree, and calculator) as seen at the computer interface. All three representations are shown concurrently.

Figure 5.5. ENCAL Interface Version 1



Not obvious whether books or shelves.

Faces

The smiling faces below each bookcase in the iconic representation indicate that each column is well-formed (refer to Chapter 4 for an explanation of well-formed columns). The smiling face next to the datatree representation shows that the numbers and operators in the "boxes" and the connecting arrows are correctly formed. The smiling face in the calculator representation informs a user that the expression above the calculator is algebraically well-formed. A sad face prompts a user to amend a representation since it is not well-formed. The feedback provided by the smiling/sad faces may not always be apparent to users. This is because they could be overlooked. In some instances, individuals may not understand the meaning of the faces. In such cases, teacher assistance will be required in order to explain: (a) how the relevant icons are not well-formed; and (b) what action needs to be taken in order to make the representation well-formed.

Carpet

A user may store books on the carpet area, and then reuse them. Whilst books are on the carpet they do not affect a calculation.

Look Inside Calculator

Users can check the calculation sequence carried out by the calculator by clicking on Look Inside. Each calculation step is shown and evaluated according to the behaviour of either the arithmetic or scientific calculator.

A subsequent pilot study was carried out based on storyboard version 1, and this is reported next.

5.3 Pilot Evaluation of ENCAL Version 1

5.3.1 Introduction

A pilot test of the ENCAL computer-based learning system was carried out over two school half-term days (08/02/99 and 10/02/99) at the Computer-Based Learning Unit, University of Leeds. The primary purpose of the pilot test was to assess the usability of the system as described in the aims below, and make subsequent changes to the interface where applicable.

5.3.1.1 Aims of the Pilot Evaluation

The pilot Evaluation had two fundamental aims. Firstly, to assess whether or not children could understand and *use* the system effectively to solve arithmetic word problems. Secondly, through user interactions, identify *learnability* problems including possible confusion due to the interface design that could hinder the learning of arithmetic.

5.3.2 Method

5.3.2.1 Design

The principal method of data collection for the pilot test was through informal direct observations of user/interface interactions. During the pilot test, the usability of the ENCAL system was evaluated using the idea of evolutionary prototyping (Dix, Finlay, Abowd, and Beale, 1993). In other words, the designed prototype was evaluated with the intention of it not being discarded afterwards, but instead serving as a basis for the next iteration of the design, assuming this to be necessary. Four aspects of usability were evaluated together with associated problems that could affect learning. The four usability aspects are listed below:

- whether users could map problem information to individual representations and vice-versa;
- whether users could map information within and between the iconic, dataflow, and calculator representations;
- whether the use of the datatree "box" metaphor and the spatial metaphor of moving book and shelf icons would be understood by users;
- whether users would take notice of all the representations, in particular the datatree.

5.3.3 Apparatus/Materials

Worksheets were provided on which were written four problem statements (see Appendix 3). A PC, monitor, and keyboard were used. The ENCAL system was programmed using Multimedia ToolBook software. For the purposes of the pilot study, only the arithmetic (i.e. four-function) calculator needed to be used during problem solving. In addition, ENCAL was set up to enable users to interact with only one of the three representations at a time. The reasoning behind this was to constrain the participants to use the representations (iconic, datatree, and calculator) in the preferred concrete-to-abstract sequence in order to facilitate understanding of problem solving. Thus, if the iconic representation was selected at set-up, it was only that representation with which the user could interact. The other two representations had a light hatching over them, but were still visible. A video camera was set up on a tripod and used to record the user and interface interaction. In addition, observational notes were taken during the pilot test by a person situated behind the participant.

5.3.4 Participants

Six children (one female and five males) took part in the pilot test - four on the first day, and two on the second day. The children were volunteers from a local primary school; the only selection criterion being that they were of the appropriate age required for the pilot test. The age of the six participants was chosen so that they ranged between nine and thirteen years. The sex and age of each child is shown in Table 5.1.

Table 5.1. Sex and Age of Participants

Sex	Age
Female	10
Male	9, 11, 11, 12, 13

Each child immediately prior to starting the pilot test was given a demonstration of the use of the three representations: iconic; dataflow; and calculator. Each child was then allowed time to practice interacting with each of the three representations. Thus, each child had the opportunity to drag and drop book and shelf icons into bookcases (iconic representation), construct a datatree (datatree representation) and enter data into the calculator (calculator representation).

5.3.5 Procedure

Each of the six children was seen individually for the pilot test. After the training period, but just before the start of the pilot test, the video camera was set to record. Four written questions were given to each child on a worksheet (see Appendix 2) at the start of the test. The questions increased in complexity from question one to question four. In addition, for solving each of the four problems, the participants were constrained to using the ENCAL representations in the following sequence: iconic; dataflow; calculator. After reading each question, the interface was used by the child in order to arrive at the required solutions. Prompts and help both with the arithmetic and use of the interface tools were provided by the experimenter where children were having difficulty. Each child recorded his/her answers on the worksheet.

5.4 Results of the Pilot Test

5.4.1 Affective Response

The children enjoyed using the system and commented on how useful they found it for learning in terms of the concrete, intermediate, and abstract representations.

5.4.2 Usability Snags

As expected, a few flaws were found in the initial design. These are referred to in the section Detailed Usability Results.

5.4.3 Success of the Approach

The pilot test showed that the ENCAL system worked in terms of participants being able to use the system and being able to solve given arithmetic problems. However, little use was made of the carpet area. This suggests that design decisions might be reconsidered during further versions of such environments. Nevertheless, the initial results are an overall reflection of the success of the design approach.

5.4.4 Detailed Usability Results

Three usability problems and their corresponding learning implications were identified with the interface as a result of the pilot test. The three problems were based on the four aspects of usability (refer to Section 5.3.2.1) which were focused on during the study. Comparisons between the youngest and the oldest participants during problem solving were also highlighted since these could have potential implications for the usability of the system. The three interface problems and the comparison between the youngest and oldest participants are described below.

5.4.5 Usability Problem One (refer to usability aspects 1, 2, and 3, Section 5.3.2.1)

The first problem was concerned with the mapping from: (a) the iconic representation to the datatree representation; (b) the datatree representation to the iconic representation; and (c) the iconic representation back to the problem statement. For example, it was not clear to all the children that three red books and three blue books on one shelf represented 3+3 as seen in the equivalent datatree (question 2), and four blue books on each of two shelves represented 2x4 (question 4) as opposed to 4x2. In addition, when constructing a datatree, the left-hand "box" represented shelves, but if the number of books was entered into this box, the corresponding iconic representation was incorrect and did not map back to the problem statement. This was a source of confusion to the participants.

5.4.6 Usability Problem two (refer to usability aspects 2, 3 and 4, Section 5.3.2.1)

The second problem was concerned with mapping from the calculator to a datatree. For example, with question 4, when children entered the expression 2x4+3x2 directly into the calculator without parentheses, the answer given was 22. Based on the problem statement, the children had previously counted the books in the iconic representation, and knew that the number of books was 14. This was a further source of confusion. The children did not check the datatree – this was largely ignored at this stage, to see if the tree construction reflected the expression needed to determine the total number of books. This problem was associated mostly with arithmetic understanding, but also with interface usability concerning datatrees.

5.4.7 Usability Problem three (refer to usability aspect 2, Section 5.3.2.1)

Referring to usability problem 2 and the expression 2x4+3x2, it was noted that if a datatree was input which did not correspond to the behaviour of the calculator, then when the calculator equals key was clicked, the datatree rebuilt itself to conform to the calculator logic. The arithmetic calculator (i.e. left-to-right convention) was selected throughout the pilot test. Therefore, when the expression 2x4+3x2 was constructed omitting parentheses using the datatree representation, then the tree rebuilt itself and gave the incorrect answer of 22 based on the left-to-right calculator logic (i.e. 2x4 = 8+3 = 11x2 = 22).

5.4.8 The Youngest and the Oldest Participants

The youngest participant (nine years of age) was familiar with parentheses, and therefore their use in the datatree and calculator representations was not alien to him. However, he had some difficulty understanding the wording of question 4. He was not sure how many books to put on each shelf. When prompted as to the meaning, he then carried on with answering the question. In addition, he had difficulty appreciating that to represent the total number of books using the datatree, the left and right subtree constructions (which represented the books and shelves in the left and right bookcases) would need to be connected by an addition operator. He had correctly constructed the datatree to represent the left and right bookcases, but omitted to connect them with an addition operator to determine the total number of books. When it was explained that an addition operator was needed, he did not know where to place the operator on the datatree. This resulted in a sad face appearing indicating that the tree was not well-formed. The concept behind the happy and sad faces had to be explained, and he then used them as a reference whilst attempting to construct the datatree. However, he needed further prompting to produce a well-formed datatree.

With question 4, the oldest participant (the thirteen-year-old) immediately realised that an addition operator was needed to find the total number of books. Furthermore, he had no trouble either: (a) in positioning the operator (i.e. between the left and right subtrees); or (b) in knowing where to make the connections to the addition operator (i.e. from the multiplication operator in each of the left and right subtrees).

5.5 Discussion and Conclusions of the Pilot Test

5.5.1 Discussion

Usability Problem One

With regard to usability problem one (refer to usability aspects 1, 2, and 3, Section 5.3.2.1) outlined in the results, confusion occurred with question 4 because the 2 represents two shelves and the 4 represents four books. However, with question 2, the number 3 plus the other number 3 both represent books and shelves are not represented at all. This result suggests that the number of shelves should be indicated in all cases. Therefore, question two may be better represented as 1x3+3 in order to avoid confusion. This consistency in arithmetic expressions could be important to facilitate learning in younger children. Having stated this, with the expression: 2x(3+5)+1x2, the 1x unnecessarily complicates the expression and would be better omitted. In other words, in one bookcase there are 8 books on each of two shelves [2x(3+5)], and in another bookcase there are 2 books on one shelf [1x2]. Therefore, in some problem statements, shelves will have to be represented in the datatree, as with 2x(3+5). However, in other cases, showing shelf numbers in the datatree will not be essential and will probably cause confusion (as with 1x2), when all that has happened in the latter case is the addition of 2 books in a bookcase. Consequently, it may be better not to show shelf numbers in the datatree when only one shelf is used.

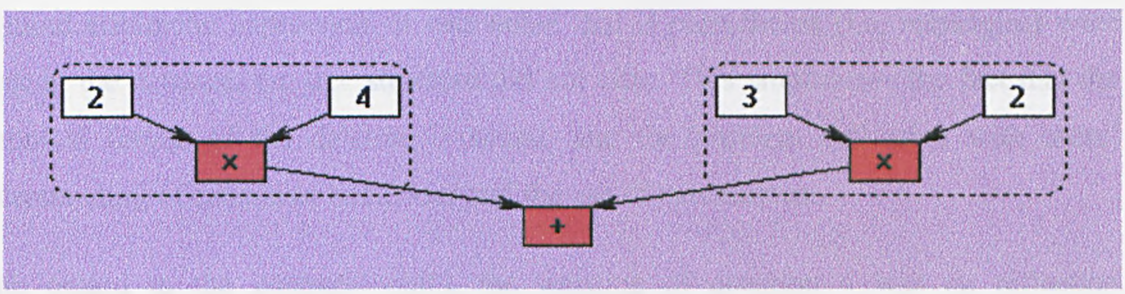
It was found that most children (apart from the 12 and 13 year olds in particular) entered the number of books into the left datatree box as opposed to the number of shelves. This was probably because the books had a more dominant presence on the interface, and the number of books were stated in questions before the number of shelves. The result was an incorrect equivalent iconic representation. So, three blue books on each of two shelves would be represented as 2x3 in the datatree and not 3x2. However, the children knew that 2x3 = 3x2, and so from their point of view the arithmetic order was not of importance, and so the only confusion was associated with the iconic representation not mapping back to and thus matching the problem statement. Essentially, the system constrained users to entering data in a sequence. From the point of view of mathematical pedagogy this was felt to be unsound, but the finding nevertheless highlighted the need to use the datatree as an intermediary check to see if the equivalent iconic representation correctly represented the problem statement.

Based on the above discussion of problem one, it was decided to alter the interface in order to avoid confusion between book and shelf representations when mapping from: (a) the icons to the datatree; (b) the datatree to the icons; and (c) the icons back to the problem statement. Thus, in the datatree representation, a menu could appear next to the datatree "boxes" which would allow users to clarify whether book or shelf numbers are to be entered. In addition, a datatree "box" which contains numbers should be appropriately coloured to represent book or shelf icons as seen in the iconic representation. It was also noted prior to the pilot test that the red book icons were the same colour as the operator keys on the calculator. Consequently, the colour of the book icons was amended in order to avoid any potential confusion.

Usability Problem Two

With regard to usability problem two (refer to usability aspects 2, 3 and 4, Section 5.3.2.1), the pilot test highlighted the facts that: (a) understanding a datatree construction is crucial to appreciating calculator behaviour; and (b) the datatree representation should not be overlooked. The thirteen-year-old child immediately noticed that the calculator answer did not match the number of books counted previously. Having entered 2x4+3x2 into the calculator, the equivalent datatree which resulted (i.e. based on left-to-right evaluation) was incorrect (i.e. the tree did not map back to the problem statement). However, the thirteen-year-old child did not scrutinise the tree to ascertain: (a) why it was structured differently to the tree he had constructed earlier to represent the problem; or (b) why the iconic representation was incorrect (i.e. it did not match the problem statement). Eventually, the tree was re-constructed with rectangles added to represent the bookcases, giving the expression shown in Figure 5.6 below.

Figure 5.6. Pilot Test Datatree Construction



Thus, 2x4 in the datatree represented the left bookcase, and 3x2 represented the right bookcase. 2x4 and 3x2 were then evaluated separately by the calculator and added together as follows: 8 (i.e. the number of books in the first bookcase) + 6 (i.e. the number of books in the second bookcase) = 14.

The discussion of Problem Two suggests that children will require a deeper understanding of how different tree structures can be constructed and how subtrees can be linked. Additionally, the children will need to know how the datatrees may be interpreted and compared to the iconic and calculator representations for checking mappings and evaluation procedures respectively. The need for more in-depth training of datatree usage would be a fundamental prerequisite if more detailed datatree construction and interpretation were to be achieved. Once users can construct and interpret datatrees effectively, an important implication of the pilot study is that if trees are checked after calculator entries, then this would be a useful guide to: (a) the behaviour of the calculator; and (b) remedial action for calculator data entry.

Usability Problem Three

With regard to usability problem three (refer to usability aspect 2, Section 5.3.2.1), on the one hand the changing tree structure was not associated with the canonical format of the datatree, but with a user's mental model of left-to-right calculator behaviour. However, on the other hand, the tree structure was correctly constructed by the user based on the problem, yet the tree structure changed when the calculator equals key was clicked. The sudden change was a source of confusion largely because the new datatree was not subsequently checked in order to ascertain the disparity between it and the required arithmetic expression. It was noted that if parentheses (i.e. rectangles) were input in the datatree, the tree structure did not alter. This emphasises the fact that the canonical format of the datatree is sound, and the problem lies more with users' understanding of calculator behaviour and the use of parentheses.

With regard to the youngest child, the wording of question 4 was on reflection unnecessarily complicated. The wording of problems would usually be carried out by teachers on work sheets and should reflect both the age and the ability of pupils. Although the ENCAL system was intended for use in the age range of 9-13 years, the pilot test results suggest that subsequent evaluations/testing of hypotheses would best involve pupils in the 11-13 age range. This is because such pupils will be sure to have been introduced to calculators and order of operations at the start of Key Stage 3 Mathematics, and in addition they will understand the more complex wording used in problems.

5.5.2 Conclusions

The participants enjoyed using the ENCAL system, and were able to interact successfully with the three representations both in terms of achieving results educationally, and handling the interface tools.

The interface needed to be modified to avoid confusion occurring with mappings between iconic representations, datatrees, and problem statements.

Datatrees tended to be ignored during calculator data entry, but this may be rectified if users are given more detailed instruction as to their construction and interpretation.

Further evaluation/testing of hypotheses would best be carried out with pupils in the upper primary, early secondary school age ranges (i.e. 11-13 years).

Since incorrect final answers using a calculator (i.e. with regard to the keying in procedures based on what a problem asked for) occurred and were not checked, hypotheses for future testing could include the comparison of calculator usage alone with the use of all three representations of the ENCAL system.

5.6 Storyboard Version 2

5.6.1 The Iconic and Datatree Representations in More Detail

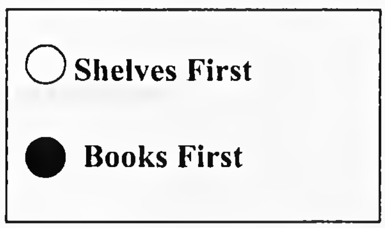
Based on the results of the pilot study, storyboard 2 was developed with the aim of improving both the usability and the educational value of the system. Only the iconic and dataflow representations have been revised in order to meet the requirements of the pilot test results. The changes to the two representations are described below.

5.6.2 The Iconic Representation Version 2

Interface Changes Which Address Problem One

Assume the expression 3+(2x4) is to be represented using the icons. A user will have the option of selecting either a group of books or shelves to be the *multiplicand* (i.e. 2). The number 4 is the *multiplier* and represents the number in the group (i.e. 4 blue/green books or 4 black shelves). It was decided to use a set-up menu that will appear prior to the representations being shown, as opposed to using a menu in each of the representations. Users may therefore select books or shelves as the multiplicand prior to interacting with any of the three representations. This is shown in Figure 5.7.

Figure 5.7. The Icons Setup Menu

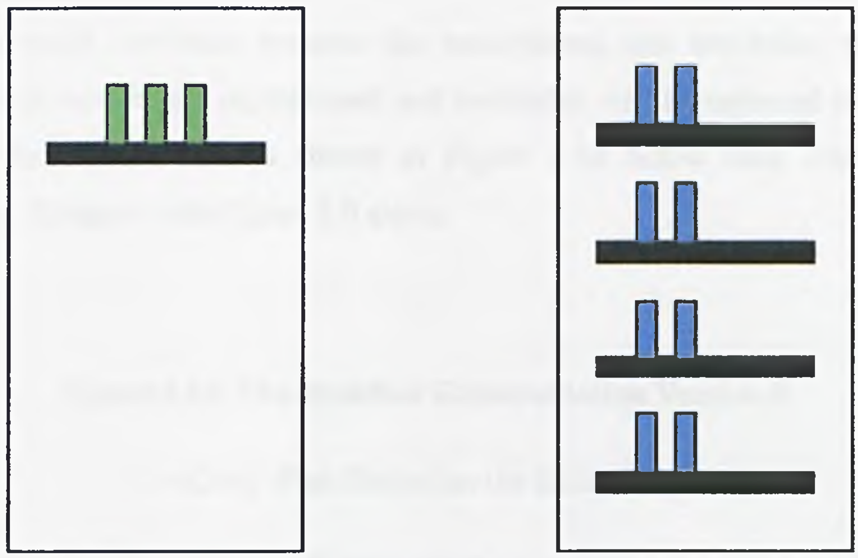


An initial start up screen will appear as shown above. A user will be required to button click either on the books button or the shelves button to select either as the multiplicand. In Figure 5.7, books have been selected. It was decided to place this menu at the start (i.e. prior to using the three representations). This is because: (a) children would be more likely to forget to select books or shelves at a later stage during problem solving; and (b) the menu in Figure 5.7 would not have to be placed in each of the three representations.

The two options of book or shelf icons as the multiplicand for the expression 3+(2x4) are shown in Figures 5.8 and 5.9.

Figure 5.8. The Iconic Representation Version 2(i)

3+(2x4), With Books as the Multiplicand

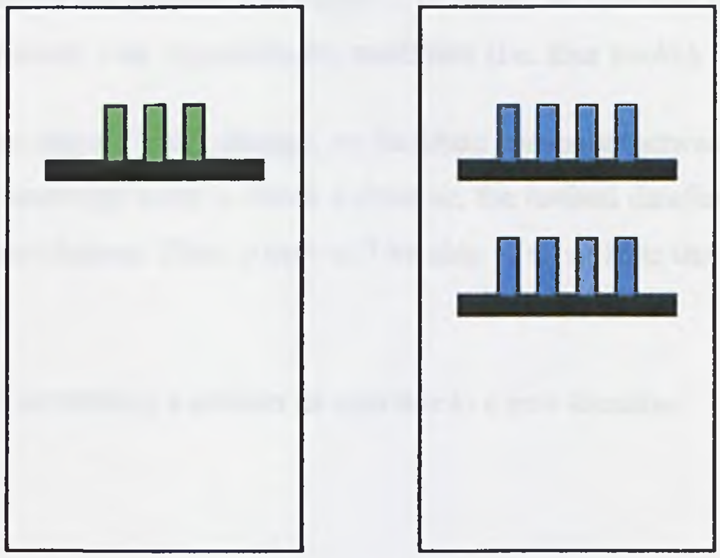


As can be seen in Figure 5.8, there are two books (multiplicand) on each of four shelves (multiplier).

Figure 5.9 below shows the reverse situation.

Figure 5.9. The Iconic Representation Version 2(ii)

3+(2x4), With Shelves as the Multiplicand



In Figure 5.9, two shelves are now the multiplicand and four books are the multiplier.

5.6.3 The Dataflow Representation Version 2

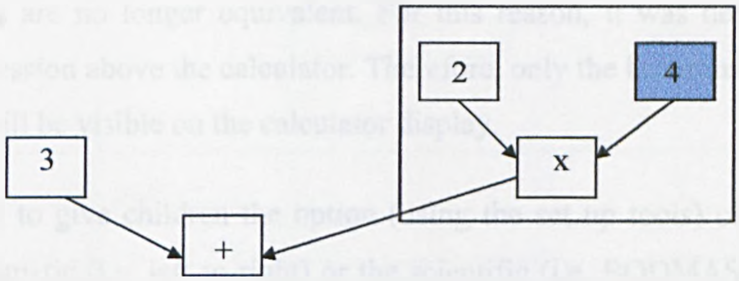
Interface Changes Which Address Problem Two

In order to avoid confusion between the multiplicand and multiplier, the datatree "boxes" which contain the multiplicand and multiplier will be coloured to match the book and shelf icons. This is shown in Figure 5.10 below with shelves as the multiplicand. Compare with Figure 5.9 above.

Figure 5.10. The Dataflow Representation Version 2

3+(2x4), With Shelves as the Multiplicand

Notice the blue box that indicates that the number contained represents the multiplier.



The "box" containing the number 2 in Figure 5.10 is the multiplicand (i.e. two shelves), and the "box" coloured blue represents the multiplier (i.e. four books).

In addition to the above "box" change, to facilitate mapping between calculator and datatree, and to encourage users to check a datatree, the revised dataflow system will be editable to a greater degree. Thus, a user will be able to do at least the following with a tree construction:

- move a "box" containing a number or operator to a new location;

- delete a connecting arrow;
- insert a new connecting arrow from a number or operator "box" to an operator "box";
- put into "boxes" numbers or operators before having to connect arrows between "boxes".

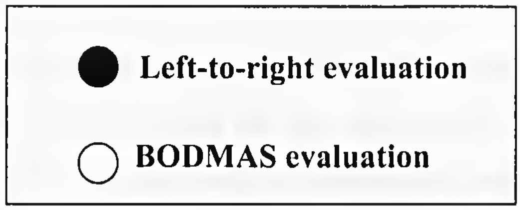
Interface Changes Which Address Problem Three

Finally, to overcome the potential confusion caused by a datatree altering its structure based on calculator logic (refer to section 5.3.3.4), it was decided that a datatree should not rebuild itself following construction if the calculator logic is changed from left to right to BODMAS (i.e. Brackets, Of, Divide, Multiply, Add, Subtract) or vice versa. The datatree will still alter when data is input to the calculator directly. In this situation, if the calculator logic is subsequently changed and then the equals key is pressed, the datatree will alter its shape. The problem then is that the calculator and datatree representations are no longer equivalent. For this reason, it was decided to hide the algebraic expression above the calculator. Therefore, only the last number entered or the final answer will be visible on the calculator display.

It was decided to give children the option (using the set up tools) of choosing to use either the arithmetic (i.e. left-to-right) or the scientific (i.e. BODMAS) calculator prior to the solving of a problem whilst using any of the three representations. This was in order to lessen the likelihood of a datatree altering its structure during user interaction. In addition, the set-up interface was redesigned to enable users to alter settings during problem solving without losing current work.

The set up menu for calculator usage is shown in Figure 5.11.

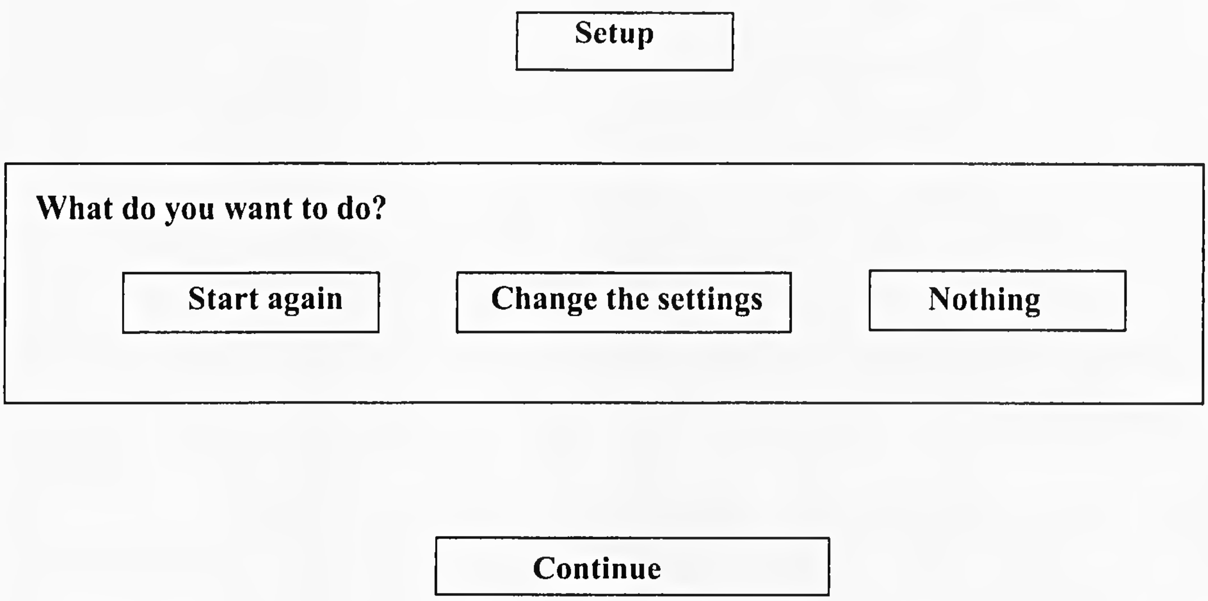
Figure 5.11. The Calculator Setup Menu



In Figure 5.11, the arithmetic (i.e. **left-to-right**) calculator has been selected.

Figure 5.12 illustrates the procedure for returning to the set up menu while problem solving.

Figure 5.12. Procedure for Returning to the Setup Menu



In Figure 5.12, the **set up** button is first clicked. If **Nothing** is selected, the system returns to the three representations (i.e. the problem-solving mode). **Continue** appears once the set up has been altered. If **Start again** is selected, a user clicks on the **Start** button that appears.

5.7 Summary

The ENCAL interface design (i.e. the computer graphics and text with which a user interacts) is an iterative process based on: (a) storyboard version 1; (b) the pilot usability evaluation and the results; and (c) storyboard version 2 (i.e. changes to storyboard version 1 based on the pilot test results).

Chapter 6 introduces the issues in evaluating the learning effectiveness of ENCAL.

Chapter 6

The Theoretical Grounds Behind ENCAL: Recapitulation

6.1 Introduction

The pilot evaluation highlighted problems of usability of the ENCAL system. These were concerned with *mappings of equivalent information between representations.* Linked to these problems are also evaluation issues associated with the learning process (i.e. the learnability of the system). In order to evaluate the learning effectiveness, we need first to restate the theoretical groundwork, and from this it may be possible to decide what observations need to be made in the final evaluation. Consequently, the following two sections explain: (a) the learning theory which underpins the design of ENCAL; and (b) the cognitive tools used with ENCAL to facilitate the learning process. Subsequently, evaluation issues concerning learning effectiveness are restated. The translatability of information is then referred to. Finally, a summary of the chapter is given.

(a) The Theory Underlying Learnability Using ENCAL

The fundamental learning theory of the computer-based environment ENCAL is based upon the Vygotskian idea that knowledge is a *process* which is constructed by a person through:

experiences;

reflection;

social interaction (e.g. discussion).

In other words, the theory states that an individual develops mental schemata which influence future thinking.

(b) Two Cognitive Tools Used to Facilitate the Learning Process

ENCAL provides the *scaffolding* (specifically, the two cognitive tools of *metaphors* and *representations*) to facilitate the learning process (i.e. the development of mental schemata) - namely, the constructive learning of concepts associated with arithmetic problem solving.

Another aspect of the learning process is that it is *situated.* Thus, a learner's experience is contained (i.e. situated) within the computer-based environment, because such situated learning provides a better (i.e. more optimal) environment for the construction of schemata. In other words, what is being learned (e.g. an evaluation sequence) is supported by the learning context (i.e. the arithmetic world of the computer-based learning environment - ENCAL).

6.2 Evaluation Issues Concerning Learning Effectiveness

Two fundamental evaluation issues of ENCAL as an educational medium are consequently concerned with the principle means by which learning is achieved with the system. These are through: (a) the use of *metaphors*; and (b) the *situated concrete to abstract translation* (i.e. mapping) of concepts between the three representations.

(a) Metaphors

Although the use of metaphors was addressed during the pilot test in terms of their usability during the solving of the given problems, a more detailed evaluation could be undertaken later on in order to assess more directly the effectiveness of the metaphors for understanding during certain aspects of problem solving. The metaphors used in ENCAL are:

iconic bookcases to simulate evaluation steps;

datatree nodes and lines to simulate operation steps;

datatree rectangles to simulate parentheses;

calculator to simulate both four-function and scientific logics;

spatial movement of entities to simulate real-life objects.

The above metaphors control the content of the computer-based environment, and they contribute to learning effectiveness.

(b) The Situated Translation of Concepts From Representations to Understanding

The translation from the symbol systems used in each representation into understanding may cause fundamental difficulties for children during the learning process. This is because the symbol systems used in each representation may or may not be readily or easily translated by individuals into meaningful mental schemata which will inform future thinking. Consequently, the use of one or more representation(s) – especially the intermediate representation (datatree), during problem solving needs to be evaluated in order to assess the translatability of information from each symbol system to understanding and thus learning. It is expected that the datatree will serve as a bridge between the concrete and the abstract. The next section describes how this is achieved.

6.3 How Translatability of Information is Achieved

The translatability of information (in particular, concepts) is made possible due to the interactive and constructive design nature of ENCAL. In other words, learners are not faced with understanding large amounts of text, but with being actively involved in constructing (i.e. building) diagrammatic representations. The computer-based graphics help users translate abstract problem text into meaningful and interactive diagrams. Information is translated from each representation (i.e. symbol system) progressively from concrete (icons) to intermediate (datatree) to abstract (calculator). It is this concrete to abstract translation process with the aid of the three graphical representations which supports understanding and thus learning.

6.4 Summary

The learning effectiveness of ENCAL is directly associated with the construction of knowledge via Vygotskian educational theory, and the interactive cognitive tools used at the interface. The use of metaphors, and the concrete to abstract movement between specifically designed representations, are the principle means by which learning is achieved. The translatability of information from the ENCAL interface to understanding is anticipated to be beneficial with the use of the intermediate datatree representation.

The next chapter describes the final evaluation including the results.

Chapter 7

The Final Evaluation

7.1 Specific Evaluation Aim

The general evaluation aim was to determine whether the three computer-based representations (iconic - concrete, datatree – intermediate, and calculator - abstract) helped both low and high attainment pupils solve mathematical problems. The iconic and calculator representations would by themselves be sufficient to achieve this, since concrete to abstract progression is evident at the interface. However, the intermediate datatree is particularly crucial since it enhances mathematical understanding by connecting the concrete and abstract extremes. Consequently, the *specific evaluation aim* was to assess the value of the datatree representation for problem solving. The experimenter (who was not the author) was naïve with respect to this specific evaluation aim.

7.2 Method

7.2.1 Design

The design was comparison-based in order to determine the effectiveness of the datatree computer-based representation. Two groups took part. One group had access to the iconic, calculator, and *datatree* representations, whereas the other group had access only to the iconic and calculator representations. User and interface interactive information was collected through direct observations which were based on participants' responses to set questions. A between–subject design was chosen because: (a) each group of participants only had to carry out the experiment once; (b) each student would only have one exposure to the experiment in either the datatree or no datatree conditions; and (c) a more realistic comparison between the two groups could ultimately be made since none of the individuals had prior exposure to ENCAL.

7.2.2 Apparatus/Materials

A desktop PC, monitor, and keyboard were used along with a modified version of the piloted ENCAL system. This was programmed using Multimedia ToolBook software. Only the arithmetic (i.e. four-function) software calculator was used during problem solving since this was sufficient to meet the evaluation aim. In addition, ENCAL was set up to enable users to interact with: (a) the iconic, datatree, and calculator representations; or (b) the iconic and calculator representations only. A video camera recorded the user and interface interactions. Observational notes were also taken during the evaluation by the experimenter. Questionnaires were provided on which were written two demonstration questions and three test problem statements (see Appendix 4). The children in group 1 did not have access to the datatree representation at the computer interface, and so their questionnaires had the datatree questions omitted. The children in group 2 had access to the datatree, and so these questions were included in their questionnaires. The test problems increased in complexity from question one to question three. Prior to the evaluation, the experimenter referred to the notes shown in Appendix 5 in order to familiarise herself with the situation since she is not the author of the thesis.

7.2.3 Participants

Twelve children took part in the evaluation. Six children were of high mathematical attainment and six children were of low mathematical attainment. Attainment was determined by the class teacher based on the children's prior accomplishment in class. Three high attainment participants and three low attainment participants were assigned to both group 1 (no datatree) and group 2 (datatree). Thus, two matched (as opposed to random) groups were produced based on their mathematical abilities. The children came from a local secondary (i.e. high) school, the only selection criteria being that they were of appropriate age and achievement required for the evaluation. The participants were chosen so that they were 12-13 years of age (i.e. 2nd year of high school), The group, sex, age and mathematical attainment (i.e. high or low) of each child is shown in Table 7.1.

Table 7.1. Group, Sex, Age, and Mathematical Attainment of Participants

Group **1 or 2**	**Number of Participants**	**Sex** **Male/Female**	**Age and Mathematical Attainment** **High (H) or Low (L)**
Group 1	1	Female	12 (L)
No Datatree	5	Male	12 (H),12 (H), 12 (H), 12 (L), 12 (L)
Group 2	3	Female	12 (H),12 (L), 13 (L)
Datatree	3	Male	13 (H)13 (H), 12 (L)

7.2.4 Procedure

The evaluation took place in the Computer Based Learning Unit at the University of Leeds. Each of the twelve children was seen individually in the same room, and each child used the same PC on which was the pre-loaded ENCAL software. The children in group 1 were constrained to interacting with the iconic and calculator representations when using ENCAL, whereas the children in group 2 had access to all three representations (iconic, datatree, and calculator). A participant's session lasted approximately thirty minutes. Prompts and help with the use of the interface tools were provided by the experimenter during both the *demonstration* and *test* questions. Throughout the *demonstration* questions pupils were: (a) shown how to solve problems using ENCAL; and (b) provided with mathematical help. However, no mathematical assistance was given during the *test* questions. The separation of mathematical and interface help are somewhat tied together in ENCAL. During the test questions the experimenter minimised mathematical help during any interventions. Instead of providing direct help with questions, she attempted to calm participants who were experiencing difficulty. However, if pupils encountered an impasse during the test questions, they were told that in a normal classroom situation they should ask the teacher or perhaps friends for help. The children not being tested were given background activities in the computer suite by their class teacher.

Once in the evaluation room, a child sat side-by-side with the experimenter and was then given either a group 1 or group 2 questionnaire. The video camera was set to record before starting the demonstration and test questions. When the demonstration was complete, a child began the three test questions. After reading a test question, ENCAL was used by a child in order to arrive at the required solution. Each child recorded answers to the test questions on his/her questionnaire. When the evaluation was finished, the video camera was switched off, and ENCAL was reset for the next participant.

7.3 Results

A coded summary of the results and the performance coding meanings from the ENCAL evaluation is shown below in Tables 7.2 and 7.3 respectively. Following the coded summary, the principal results within this are described. Then several tables highlight the major findings. All the results are depicted with respect to the evaluation aim (i.e. to determine the helpfulness of the datatree). Appendix 6 shows examples of the coding scheme used whilst classifying behaviours.

Table 7.2. Coded Summary of the Results

Coded Summary of Results From the ENCAL Evaluation										
					Calculator Answer Correct (CT) or Incorrect (IT)			*Performance Coding for the Three Test Questions*		
Participants' Initials	*Attainment*	*Sex*	*Age*	*Group*	*Q1*	*Q2*	*Q3*	*Q1*	*Q2*	*Q3*
R.L.	LOW	M	12	1	CT	IT	IT	QCT	UP/OP/BP/ICU	OP/BP
D.P.	HIGH	M	12	1	CT	CT	IT	MRI	OP/BP	OP/BP
D.F.	HIGH	M	12	1	CT	IT	IT	QCT	UP/BP/MRC	UP/BP/ICU
C.S.	LOW	M	12	1	CT	CT	IT	WT	WT	WNB/MUW/ICU
L.A.	LOW	F	12	1	IT	IT	IT	QNA/WNB/OP	QNA/WNB/OP/BP	QNA/WNB/OP/BP
A.G.	HIGH	M	12	1	CT	CT	CT	QCT	OP/BP/WT	OP/BP/WT
H.P.	LOW	F	12	2	CT	IT	CT	QCT	MRC/FW/WT	FW/ICU
T.M.	LOW	M	12	2	CT	CT	CT	QCT	QCT	WNB
R.W.	HIGH	F	12	2	CT	CT	IT	QCT	QCT	BP
N.J.	HIGH	M	13	2	CT	CT	CT	QCT	QCT	QCT
L.D.	HIGH	M	13	2	CT	CT	CT	QCT	QCT	MRD
E.B.	LOW	F	13	2	CT	IT*	IT	QCT	WNB	WNB/ICU/BP/ID

Table 7.3. Performance Coding Meanings

Performance Coding Meanings		
QCT	**Question Correct**	A question was answered correctly using calculator and icons with appropriate working.
UP	**Understanding Problem**	Calculator answer wrong, but iconic answer right indicating failure to understand problem, maths or calculator.
OP	**Operator Problem**	The multiplication or addition operators were omitted or not used appropriately.
BP	**Brackets Problem**	Brackets were either not used or were used incorrectly.
ID	**Interpreting Datatree**	Users incorrectly translate bracket information from a datatree to an expression.
WT	**Wrong Thinking**	A person gives the correct answer but has not used the calculator “look inside” information for the working.
QNA	**Question Not Answered**	An individual has not addressed a problem statement.
WNB	**Wrong Number of Books**	A wrong answer is given due to books mis-counted or incorrect books on shelves.
MUW	**Made-Up Working**	Working has been made-up to fit an answer.
MRC	**Mis-Read Calculator**	An incorrect calculator answer was written down despite the working being correct.
MRI	**Mis-Read Information**	The final answer is correct, but a user has mis-read the calculator “look inside” information for the working.
FW	**Fragments of Working**	Fragments of working visible on screen from the datatree.
ICU	**Incomplete Calculator Use**	Got part way through a question, but never finished.
MRD	**Mis-Read Datatree**	Some datatree number, operator, or bracket was read incorrectly, but most of it was correct.
*	**Slip**	Incorrect only because of a slip (i.e. in number, operator, or problem with reading).

Possible Gradings	**Correct, correct by luck.**
	Incorrect because of a slip. **Incorrect because of a bracket misplacement.** **Incorrect where a user had significant mathematical problems and/or difficulty in understanding questions.**

When considering high and low attainment pupils together, the summary of coded results referred to in Table 7.2, shows that the two most frequent performance errors in group 1 (no datatree) per participant and per question were: OP, and BP. The error UP was also present in questions two and three, but to a lesser extent. In comparison, group 2 (datatree), had no OP or UP performance errors, and only two BP performance errors with question 3. Two pupils from each group had ICU errors with question three. In group 1, four out of the six participants had performance errors with question one, but all the participants had errors with questions two and three. By comparison, none of the participants in group 2 had a performance error with question one. Two pupils had errors with question two, and one of the six pupils achieved a completely correct answer with question three.

The number of wholly correct answers (i.e. using both icons and calculator) for high and low attainment participants in groups 1 and 2 is shown in Table 7.4.

Table 7.4. Comparison Between Groups (With or Without the Datatree) of Wholly Correct Answers for High and Low Attainment Pupils

	Wholly Correct Answers High Attainment Pupils	**Wholly Correct Answers Low Attainment Pupils**
Group 1 (No Datatree)	2	1
Group 2 (Datatree)	6	4

Table 7.4 indicates that both high and low attainment pupils in the datatree group obtained a greater number of completely correct answers compared to pupils who did not use the datatree.

The number of wholly correct answers for each question for high and low attainment participants in groups 1 and 2 is shown in Table 7.5.

Table 7.5. Comparison Between Groups (With or Without the Datatree) of Wholly Correct Answers per Question for High and Low Attainment Pupils

	Wholly Correct Answers High Attainment Pupils			Wholly Correct Answers Low Attainment Pupils		
The Three Test Questions	**Q1**	**Q2**	**Q3**	**Q1**	**Q2**	**Q3**
Group 1 (No Datatree)	2	0	0	1	0	0
Group 2 (Datatree)	3	3	1	3	1	0

When analysing each question, Table 7.5 shows that question 2 was answered entirely correctly (i.e. using both icons and calculator) by all three high attainment pupils and by one low attainment pupil in the datatree group. None of the pupils in group 1 (no datatree) answered question 2 correctly. Question 3 (the most difficult) was answered correctly by one high attainment participant in the datatree group, however no participant in the low attainment datatree group achieved a wholly correct answer. In addition, none of the pupils in the no datatree group (either high or low attainment) answered this question correctly.

Table 7.6 displays the overall number of correct and incorrect calculator answers for high and low attainment pupils in each group.

Table 7.6. Comparison Between Groups (With or Without the Datatree) of Correct Calculator Answers for High and Low Attainment Pupils

	Correct Calculator Answers High Attainment Pupils	Correct Calculator Answers Low Attainment Pupils
Group 1 (No Datatree)	6	3
Group 2 (Datatree)	8	6

Table 7.6 indicates that high and low attainment pupils in the datatree group performed better in terms of correct calculator answers compared to the no datatree group. The low attainment pupils in the datatree group showed a markedly higher number of correct answers compared to pupils in the no datatree group.

The number of correct calculator answers for each question for high and low attainment participants in groups 1 and 2 is shown in Table 7.7.

Table 7.7. Comparison Between Groups (With or Without the Datatree) of Correct Calculator Answers per Question for High and Low Attainment Pupils

	Correct Calculator Answers High Attainment Pupils			Correct Calculator Answers Low Attainment Pupils		
The Three Test Questions	**Q1**	**Q2**	**Q3**	**Q1**	**Q2**	**Q3**
Group 1 (No Datatree)	3	2	1	2	1	0
Group 2 (Datatree)	3	3	2	3	1, 1*	2

A noteworthy result in Table 7.7 is that two low attainment pupils in the datatree group answered question three correctly with the calculator, compared with zero low attainment pupils in the no datatree group. Two high attainment pupils in the datatree group answered question three correctly. In addition, all three high attainment pupils in the datatree group answered questions one and two correctly. All three pupils in the low attainment datatree group answered question one correctly. Also, one person answered question two correctly, and one other pupil would have given a correct answer but for a slip with a number.

7.4 Summary of Results

When considering high and low attainment pupils collectively, the coded summary of results reveal that group 1 (no datatree) had numerous performance errors in all three questions, notably OP, BP, and UP. However, group 2 (datatree) had no OP or UP errors, and only one BP error in question 3. With regard to the tables of results, the high and low attainment pupils in the datatree group obtained: (a) a greater number of completely correct answers (i.e using both the iconic and the calculator representations); and (b) a superior number of correct calculator answers. In addition, the low attainment pupils in the datatree group achieved notably more correct calculator answers for question 3 (the hardest question) compared to the low attainment pupils in the no datatree group.

A summary of these results is shown as two tables in Appendix 7.

The next chapter discusses the implications of the results.

Chapter 8

Analysis of Errors

8.1 Introduction

Although both high and low attainment pupils from groups 1 and 2 demonstrated beneficial performance using ENCAL, some errors nevertheless occurred. In this chapter we consider the errors in detail. Performance errors with operators and brackets were particularly evident with group 1 (no datatree). In particular, group 1 participants found difficulty with questions two and three, although a single pupil had similar problems with question one. Initially therefore, analysis of the operator problems (OP) and brackets problems (BP) from group 1 are described below. There were no operator errors in group 2 (datatree). Following this, the brackets problems of the only two pupils in group 2 with this error are examined. The feedback provided by the experimenter is then considered in order to gain a teacher's/researcher's point of view in contrast to the participants' perspectives.

8.2 Analysis of Operator Problems

Two low and two high attainment pupils (L.A., R.L. and D.P., A.G. respectively) from group 1 (no datatree) had operator problems with questions two and three when using the software calculator. In addition, a low attainment pupil found difficulty with question one. Analysis of just one participant from the high attainment pupils was carried out since their errors were similar. Thus, looking at both pupils would not have added to the discussion. Participant D.P. was chosen, because as well as the operator uncertainty, he neglected to answer the second half of question three. With regard to the two low attainment pupils, L.A. used an incorrect operator with question one, and R.L. missed out an operator in question two. Thus, the mathematical mistakes made by L.A. (Q1), R.L. (Q2) and D.P. (Q3) are discussed below.

L.A. (Low Attainment Pupil Group 1) Q1

Prior to using the calculator, L.A. had operator difficulties. Instead of counting (as the question asked) the two green books and three blue books (i.e. 2 + 3 = 5) which she had positioned correctly in the iconic representation, she multiplied the two sets of books and obtained an answer of 2 x 3 = 6. The experimenter said: "What does it say then?" When using the calculator to determine the total number of books in the bookcase, she pressed the keys 6 x 1 =. Consequently, six books on one shelf appeared in the iconic representation.

L.A. appeared to guess what to do which suggests she did not have a clear mental model of the problem. The visualisation of the problem was not apparent to her. Why did she use multiplication rather than addition? Two observations may provide a partial answer. Firstly, the very first demonstration question involved multiplication only, and the second question involved addition and multiplication. So since both had multiplication, she could have felt that this problem ought to have multiplication too. Secondly, added to this is the observation by the experimenter that this particular participant went into a state of panic when confronted by the research situation. She was more evidently panic stricken during the ordeal of the test questions, but it would appear that even in the demonstration questions she was in a similar state. During the demonstration, the experimenter attempted to put this person at ease. For example, she said: "Ok, very good. All it wants you to do is write that bit down there. Press CA to clear the screen and then turn over and do another little one." It is therefore suggested that this child was responding to the situation in the best way she knew how, but she was unable to muster the cognitive resources to manage the task. In this case, even a much simpler interface may have induced the same response, and so it would seem this participant adds little to our understanding of the cognitive effects of ENCAL, other than to indicate the importance of participants being at ease with the context. More practice and guidance may well have helped L.A. overcome her difficulties.

The iconic and calculator representations by themselves did not appear to help L.A. choose the correct operator, which might indicate insufficient feedback from ENCAL. In fact, the experimenter ended up by saying: “Just write it down, because that’s alright.” In other words, she attempted to put the pupil at ease. A primary problem associated with this issue is therefore the design of ENCAL. Currently, the system does not allow users to look back and compare an earlier iconic version with a newer form. In other words, a pupil has to remember his/her initial input because any later version completely replaces what already exists. Also related to this issue is the facility for appropriate teacher feedback, as opposed to just relying on ENCAL’s feedback. A teacher’s input would have shown L.A. her iconic error. If L.A.’s iconic representation and the teacher’s version were then enabled to be compared using the computer-based interface, L.A. would have been better informed prior to using the calculator. It must also be remembered that the system was not designed to be completely intelligent. Rather, a supporting datatree is used to link the iconic and calculator extremes, this and the pupil-teacher *iconic comparisons* would have helped enhance L.A.’s performance.

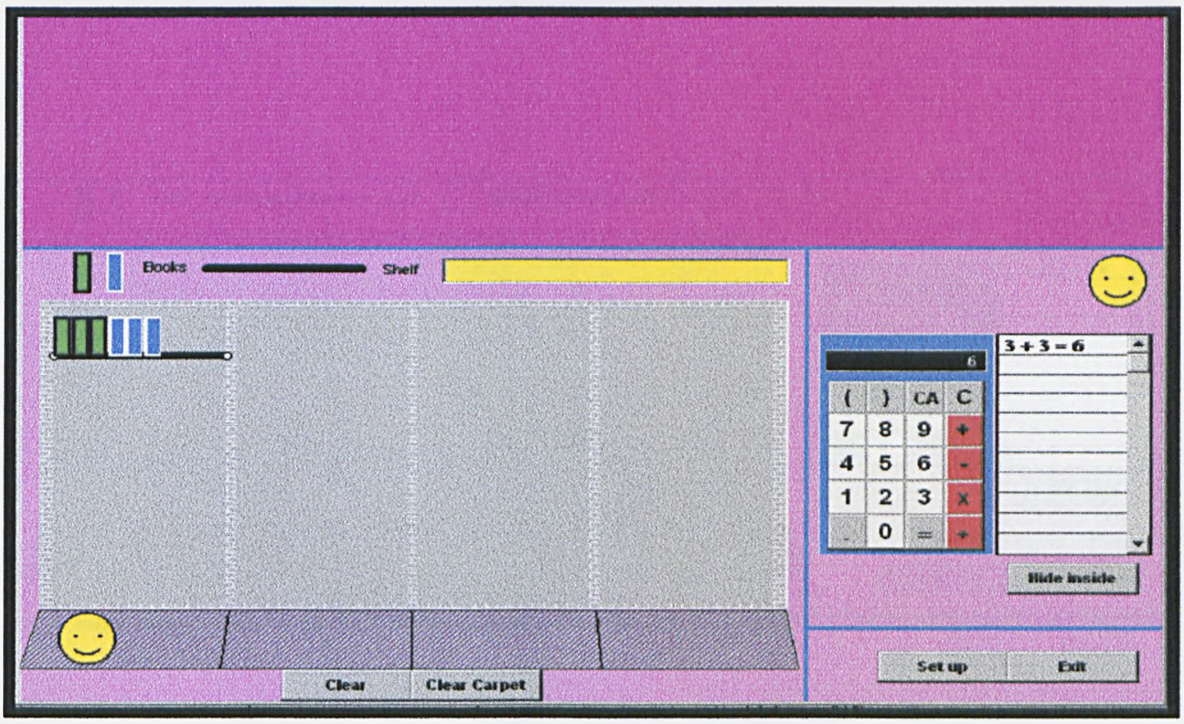
Significance of This Result. Having access to previous and current iconic representations as well as the more complex, but helpful datatree, might further assist understanding especially with low attainment pupils.

R.L. (Low Attainment Pupil Group 1) Q2

Part (b) of question 2 requires use of the calculator to solve the problem. Pupil R.L. correctly answered part (a) with the icons, but stated on the question sheet reference part (b) “I got a bit confused”. This indicates that he did not understand one or more of the following: how to solve question 2 (b); the mathematics; or the calculator.

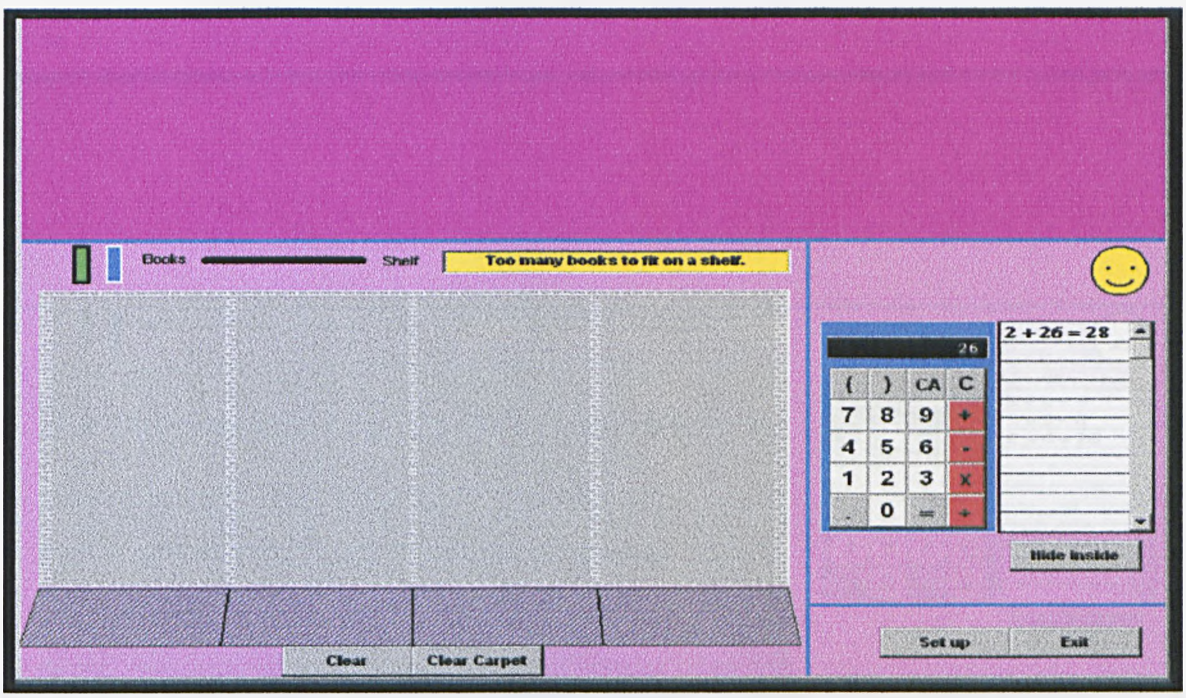
The formal expression needed prior to data input to the calculator was: 3 + 3 + (2 x 2), however this was not apparent to R.L. from the iconic representation or from his own mental model of the problem. He entered the following calculator key presses: (3 + 3 =). The resulting ENCAL screen is shown in Figure 8.1. The blank section at the top of the display hides the equivalent datatree representation. Also shown is the "Look inside" information – i.e. the calculator steps which R.L. has carried out so far.

Figure 8.1. R.L. Q2 (b)



Directly following (3 + 3 =), R.L. entered 2 + 26. He missed out the operator between the 2 and the 6, consequently the calculator display showed 26, and the message "Too many books to fit on a shelf" appeared in the iconic representation (the maximum number of books allowed on a shelf is nine). Therefore, the iconic representation disappeared because the icons were no longer well-formed[3]. Equivalence between representations was thus lost, and this could have caused confusion. To further add to the confusion, as soon as 2 + 26 was input, 3 + 3 = 6 disappeared from the "Look inside" information, because the bracket was placed after the equals sign. Thus, 2 + 26 = 28 only is shown (see Figure 8.2.).

Figure 8.2. R.L. Q2 (b) Continued



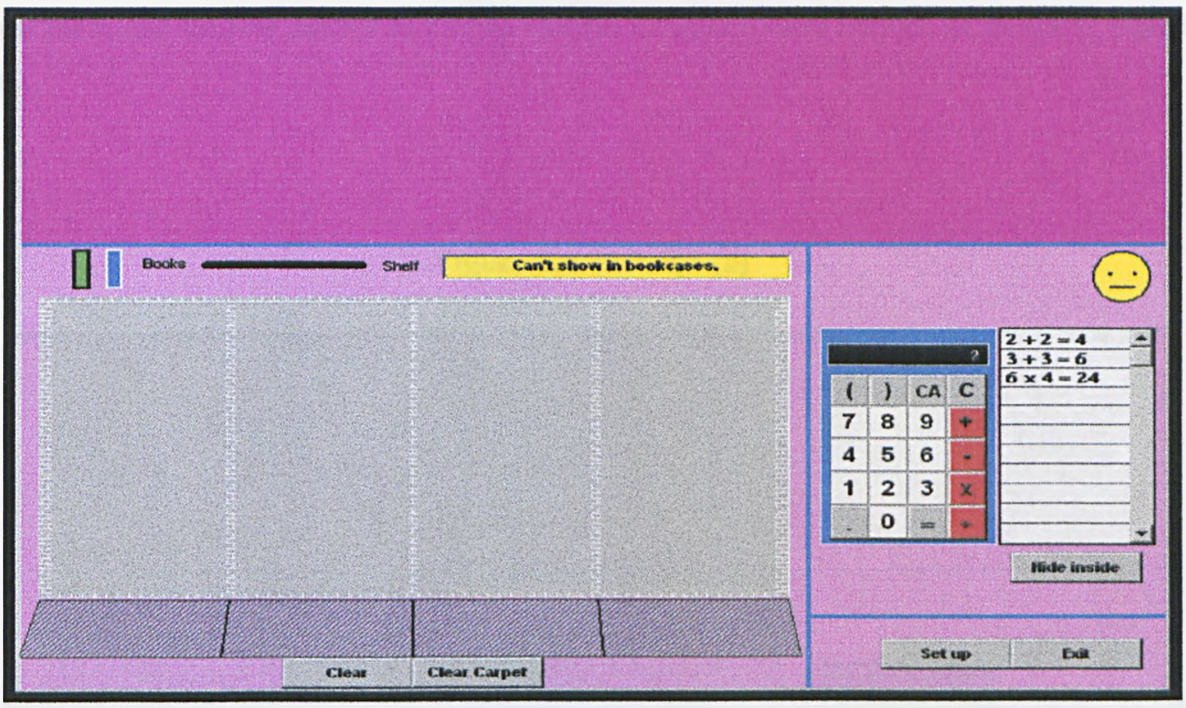
Without ENCAL, R.L. would most likely have been unaware of his operator error. The ensuing uncertainty at least caused him to think about the situation he was now in with regard to the question, the mathematics, and ENCAL. The entry of number 26 presented a

---

[3] Chapter 4 explains what is meant by the term well-formed.

problem for R.L. in particular, because it showed he did not know how to enter data into the calculator. This appears to be the reason why the calculator answer was grossly wrong, but the iconic answer was correct. After pressing the CA (Clear All) key, he then entered: 3 + 3 (2 + 2, but failed to close the bracket (this is a general problem). The message "Can't show in bookcases" appeared because R.L. did not place an operator between the steps 3 + 3 and 2 + 2. This is shown in Figure 8.3.

Figure 8.3. R.L. Q2 (b) Continued



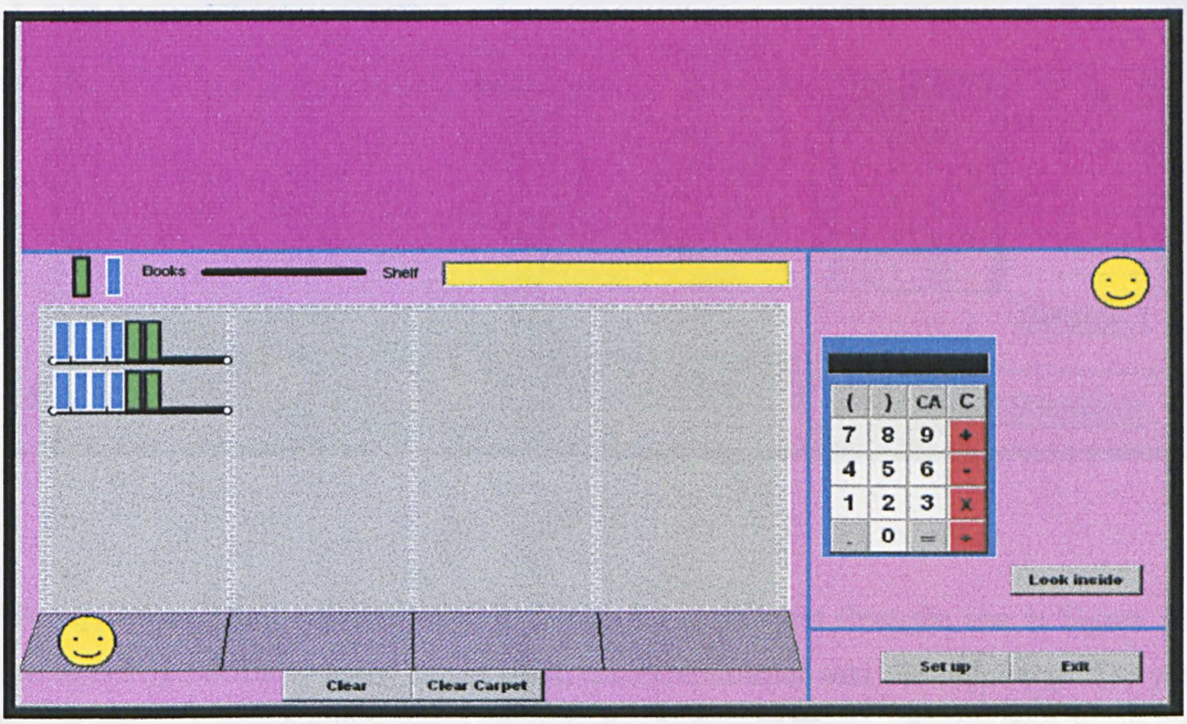
The reason for omitting the closing bracket may have been due to the experimenter's verbal intervention who realised that R.L. was struggling. She said: "It doesn't matter…do you want to leave that one and go on to the next one?" The final calculator steps which R.L. wrote on the question sheet from the "Look Inside" calculator information after completing part (b) were: 2 + 2 = 4; 3 + 3 = 6; and 6 x 4 = 24. The calculator assumed the product of the two steps was required, i.e. 3 + 3 (2 + 2 = 6 x 4 = 24, due to the bracket input by R.L. which precedes the first 2.

Significance of This Result. Losing the iconic view creates confusion and leads to misunderstanding because current screen information and equivalence between representations disappears. This result accentuates the need for possible future design improvements, and teacher feedback in order to promote learning.

D.P. (High Attainment Pupil Group 1) Q3

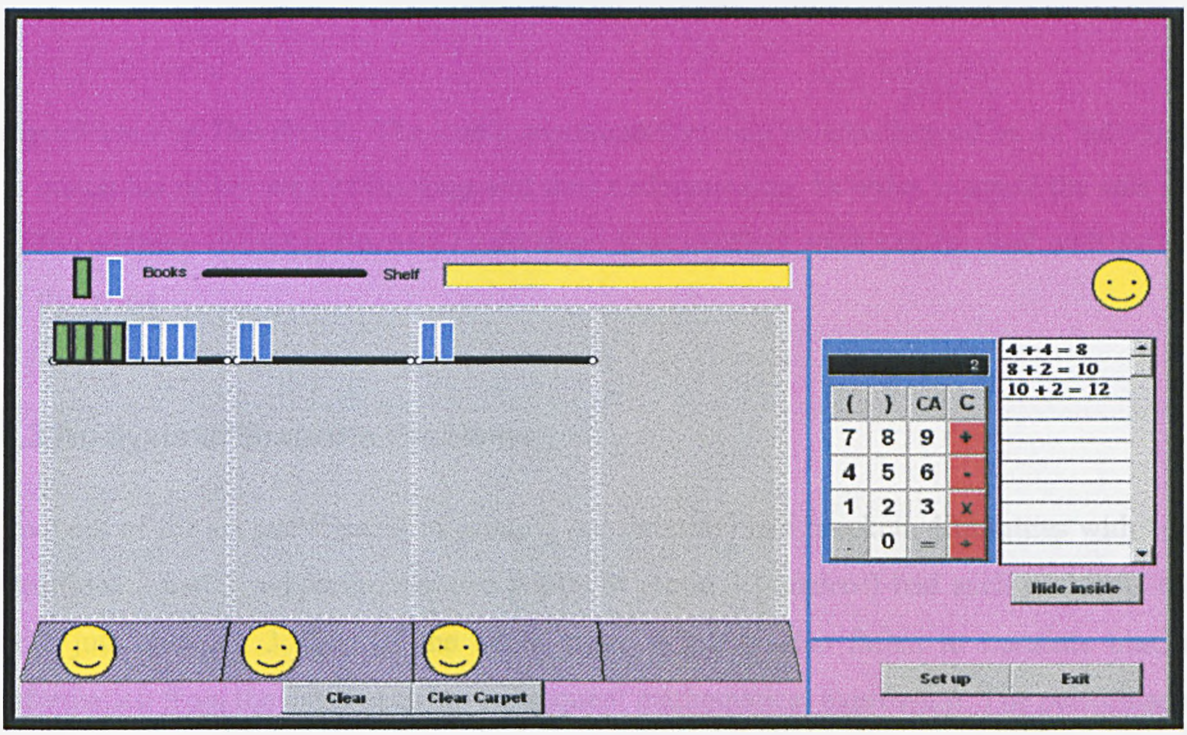
Only half the question was answered (i.e. the contents of the first bookcase) for both the iconic and the calculator representations. Both his partly completed iconic representation and his corresponding answer of 12 (the number of books) were correct, as shown in Figure 8.4.

Figure 8.4. D.P. Q3 (a)



The video tape showed that D.P. went straight to the calculator after completing the icons for the first bookcase. He entered the following expression: 4 + 4 + 2 + 2. This was obtained as a result of D.P. *transforming* the iconic representation to suit his understanding. In other words, he added the four blue books on the first and second shelves (i.e. 4 + 4), and then he added the two green books from the two shelves (i.e. 2 + 2). D.P. then added the expressions to give 4 + 4 + 2 + 2. See Figure 8.5.

Figure 8.5. D.P. Q3 (b)



The equivalent iconic representation in Figure 8.5 is different from the one in Figure 8.4 due to D.P.'s calculator entry. In response, the experimenter said the following: "Do you have a calculator at school? Is that a BODMAS calculator?" The pupil said: "I'm not sure."

The experimenter then asked: "Is it one of these scientific ones?" The pupil answered: "Yes." The experimenter then commented: "That's really interesting, thanks." The answer is correct, and is a demonstration of his cognitive resources being used in a way which is appropriate to his mathematical mental model of the problem. His strategy negates the use of multiplication – i.e. 4 + 2 x 2, but yet it expresses creative thinking with the addition operator through his transformation of the iconic representation. With suitable teacher intervention, he could see how: (a) multiplication may be used as well; and (b) the answer can be obtained in one bookcase instead of three. D.P. wrote down the "Look inside" calculator information as shown in Figure 8.5. After completing this part, D.P. pressed the Clear All (CA) button and did no further work with the question - specifically addressing the contents of the second bookcase.

Significance of This Result. The iconic representation may be transformed by an individual to match his or her way of thinking and thus understanding. In order to avoid the sole use of the addition operator, teachers could show how multiplication can be used with the aid of the icons.

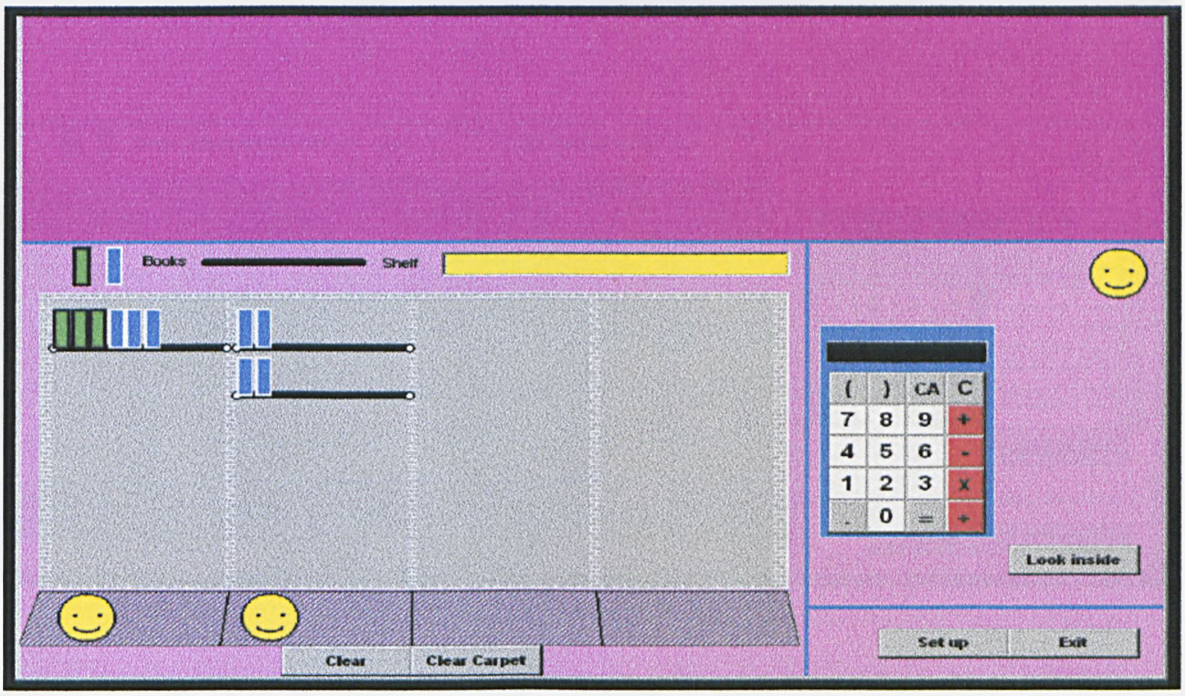
8.3 Analysis of Brackets Problems

Five out of the six participants in group 1 (no datatree) had brackets difficulties with both questions 2 and 3, whereas only two pupils in group 2 (datatree) had such problems and with just question 3. One of the latter group 2 pupils also failed to translate bracket information from the datatree to her subsequent mathematical expression. The performance errors of two pupils from each of group 1 and group 2 are analysed next (D.P. Q2, R.L. Q3, R.W. Q3, and E.B. Q3 respectively).

D.P. (High Attainment Pupil Group 1) Q2

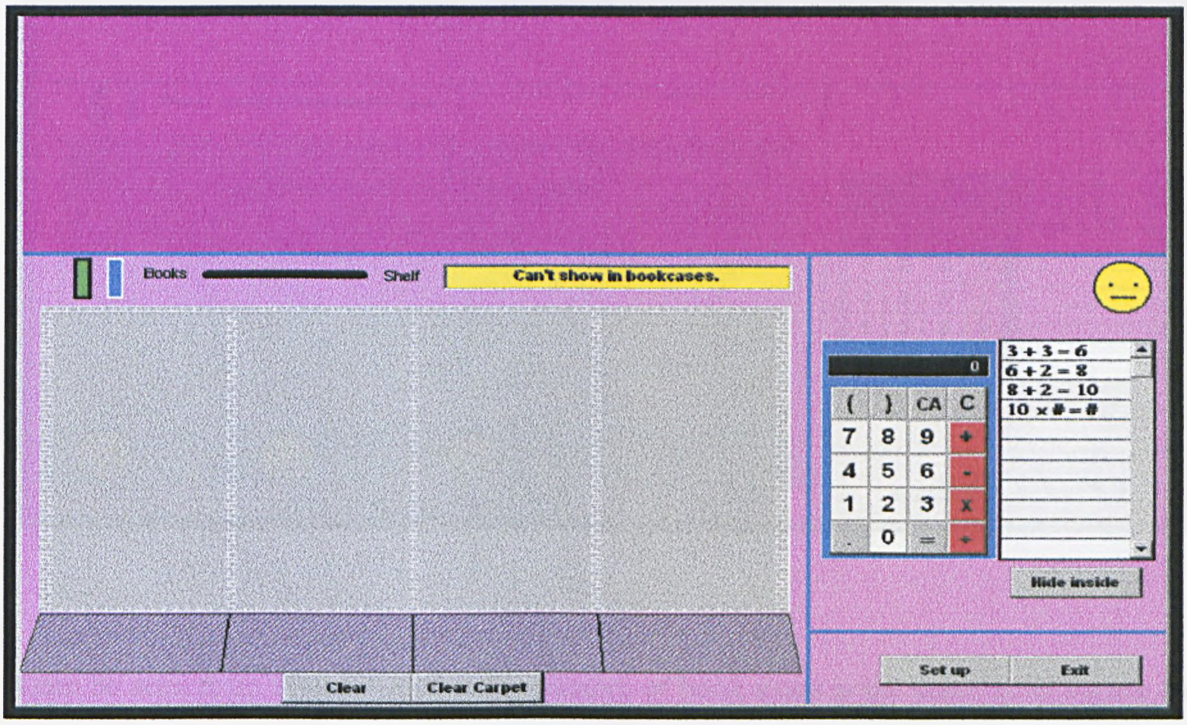
D.P.'s iconic representation of the problem statement was correct, and this is depicted in Figure 8.6.

Figure 8.6. D.P. Q2



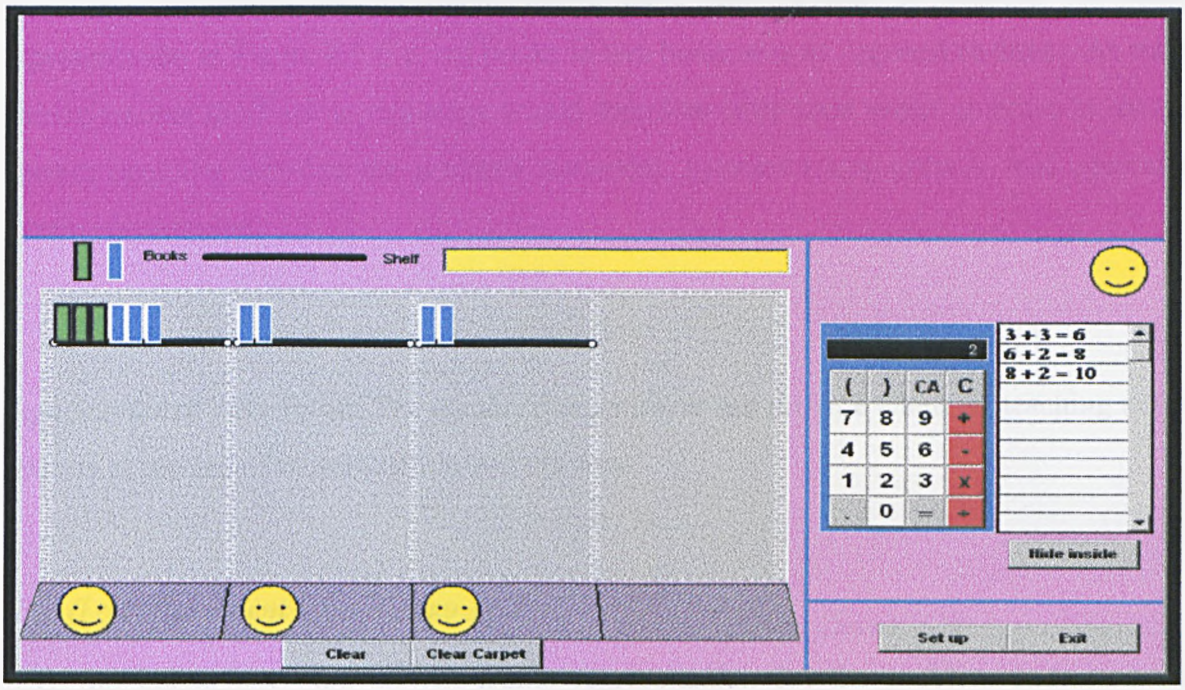
However, when he used the calculator, D.P. initially entered the following: 3 + 3 + 2 + 2 x, as shown in Figure 8.7.

Figure 8.7. D.P. Q2



D.P. comprehended that this was wrong since there was no equivalent iconic representation. The screen message "Can't show in bookcases" appeared when he inappropriately used the multiplication operator. The experimenter said "Don't worry about that". He then tried again, and entered 3 + 3 + 2 + 2. The resulting equivalent iconic display did not correspond with Figure 8.6, but he counted the books and wrote down the correct answer of 10 on the question sheet. Figure 8.8 shows the calculator steps and the equivalent iconic representation which resulted from D.P.'s calculator entry.

Figure 8.8. D.P. Q2 Continued



It can be seen from Figure 8.8, that the number of book icons is the same as in Figure 8.6, thus enabling the right answer to be achieved. However, the iconic structure in Figure 8.8 is different from that in Figure 8.6, which indicates that a disparate equivalence occurred due to D.P.'s calculator procedure. It is unclear as to whether D.P. noticed the difference in iconic structure between his calculator procedure (Figure 8.8) and his initial iconic representation (Figure 8.6). If he did not realise there was a discrepancy, then teacher input would need to refer to the question and point out that the icons need to be placed in two bookcases instead of three. To achieve this, the teacher would need to explain the use of the multiplication operator. However, if D.P. was aware of the inconsistency, then the teacher would not need to refer back to the problem, but draw his attention to the use of the multiplication operator and brackets. Thus, the focus would be less on understanding the problem requirement, but more on achieving the correct mathematical procedure.

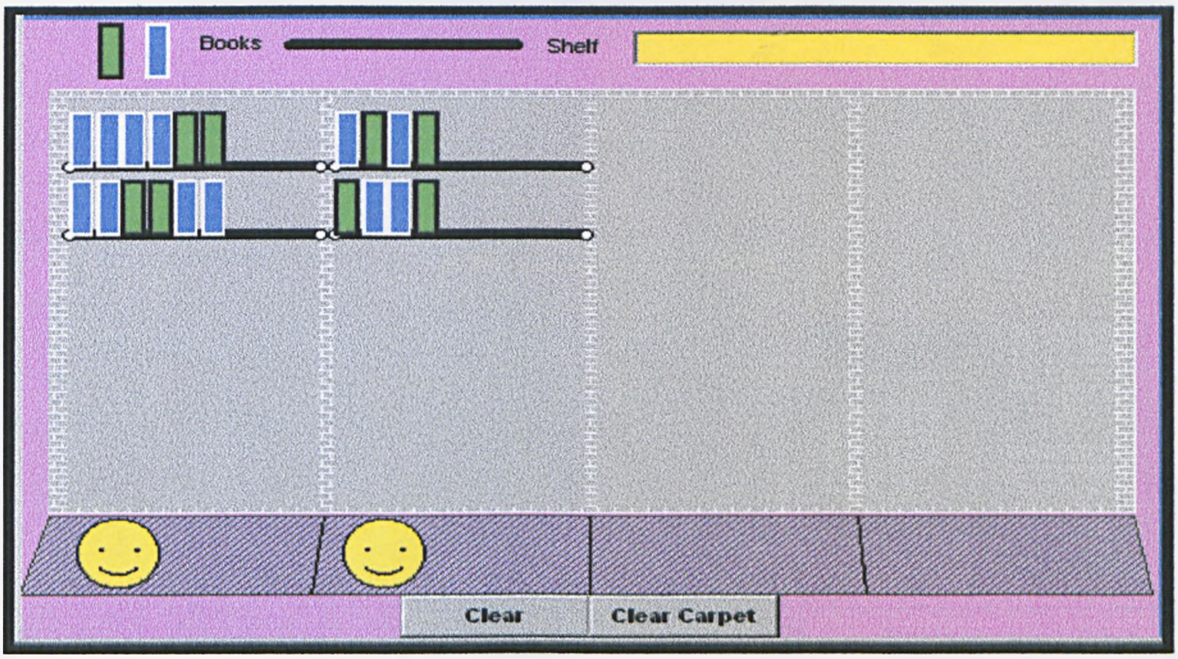
Again, D.P. only used the addition operator. He also failed to exploit multiplication and brackets, appropriate use of which would have enabled him to achieve the iconic representation in Figure 8.6 (i.e. ten books in two bookcases as opposed to three). As stated above, teacher feedback at this stage would therefore have been useful in order to achieve correct mapping between the problem statement, calculator key presses, and equivalent iconic construction.

Significance of This Result. The iconic representation can provide a helpful teaching aid for lessons in the use of multiplication and brackets.

R.L. (Low Attainment Pupil Group 1) Q3

R.L. did not achieve the correct iconic representation for the problem statement. His dragging of books was unsystematic which resulted in book colours being mixed. This could have caused confusion, since the number of blue books was incorrect in the second bookcase (i.e. two instead of three). In addition, he placed two instead of three shelves in the second bookcase. Consequently, R.L. failed to translate the problem statement to the iconic representation. In other words, his structural mapping was muddled. See Figure 8.9.

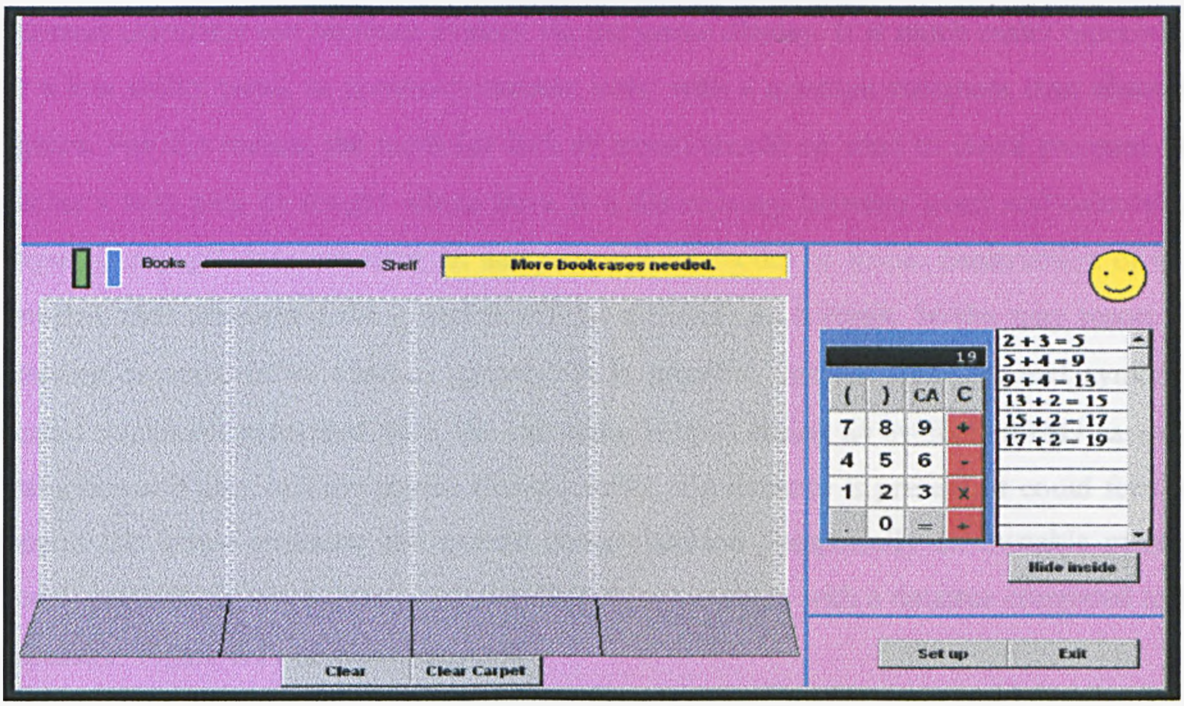
Figure 8.9. R.L.'s Iconic Representation for Q3



R.L. wrote down the answer of 20 for the number of books in the bookcases, and then pressed Clear All (CA) and used the calculator to answer the next part of the question as instructed on the question sheet. He selected 2 on the key pad after which the equivalent iconic representation displayed two books on a shelf. He discontinued this approach, and cleared all. He then pressed 1 x 2 + 4 x 2 + + 2, followed by CA. R.L. continued by selecting: 2 + 3 =; 4 + 4 + 2 + 2 + 2 =.

As R.L. input the mathematical sequence of this problem, the equivalent iconic display generated by ENCAL was different from his own iconic display, and this caused more confusion. In addition, R.L. made no attempt to use the brackets from the calculator because they were beyond his mathematics knowledge. Moreover, he was not confident with multiplication because his answers involved only the addition operator. R.L.'s calculator working for question three is shown in Figure 8.10.

Figure 8.10. R.L.'s Calculator Representation for Q3



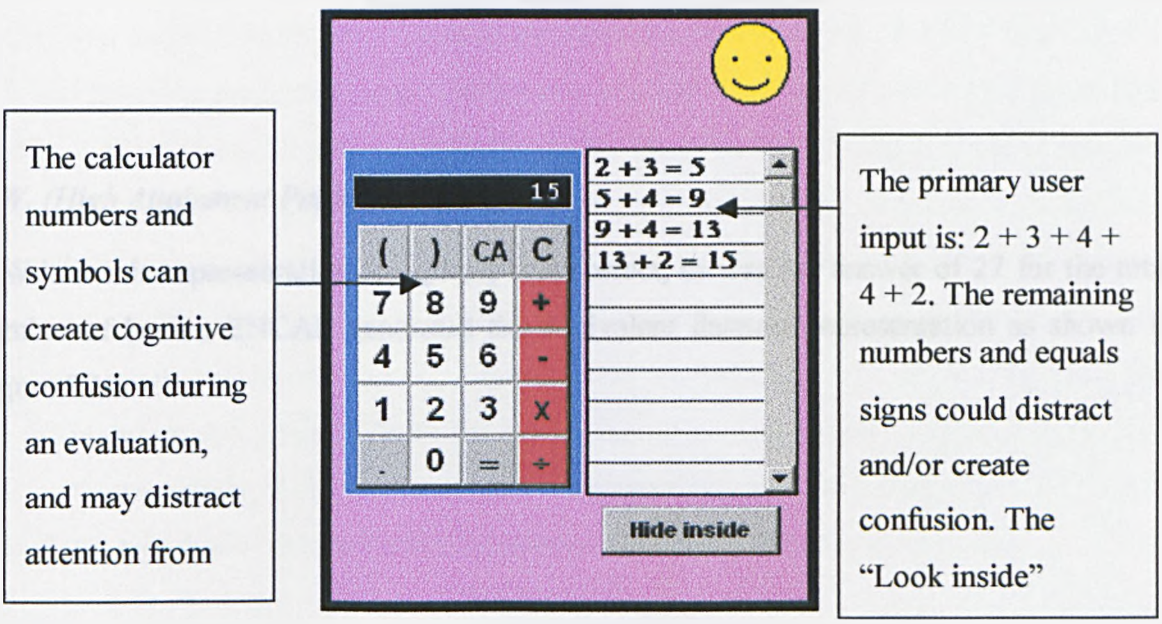
There is a distinct mismatch between R.L.'s iconic representation in Figure 8.9 and his subsequent calculator input. Figure 8.10 does not have an equivalent iconic display, because more bookcases would be needed (as indicated in Figure 8.10) to accommodate the latter two steps, 15 + 2 and 17 + 2. Despite no iconic representation after the fourth step (i.e. 13 + 2 = 15), R.L. carried on with the calculator as shown above. It might be that R.L. continued to use the calculator so as to arrive at an answer, in this case 19, which was as close as possible to his iconic answer of 20. Perhaps therefore, R.L. made up the calculator working to compensate for his lack of understanding of the problem. In addition, his iconic representation in Figure 8.9 does not show three books of one colour grouped together. Therefore, he may have read the problem again and formed a mental model of the books and shelves in the two bookcases, and as a result he then entered the calculator numbers shown in Figure 8.10 – i.e. 2, 3, 4, 4, 2, 2, 2. There could be other reasons for his working, but insufficient evidence is available from the results.

Significance of This Result.

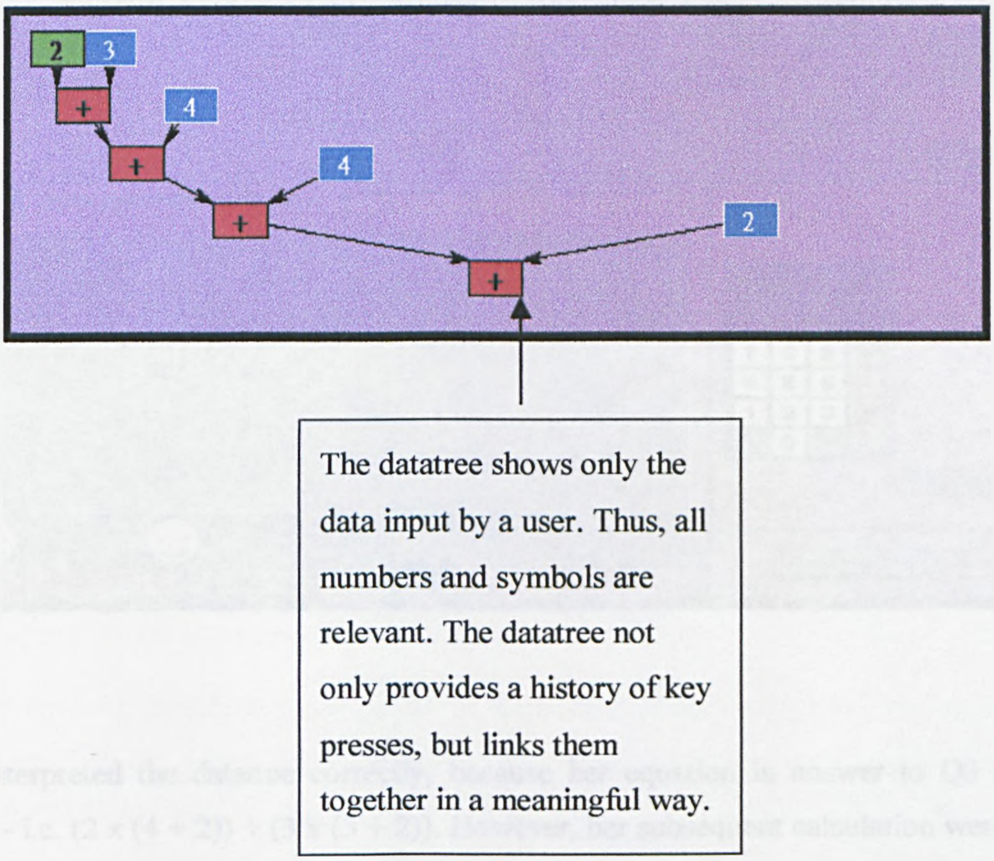
Incorrect calculator use becomes evident via the iconic display. It is much easier especially for a low ability pupil, to compare concrete icons with a question statement than abstract algebra. For this reason, the icons (or lack of icons) should be used to check progress. A teacher's help may be sought where there is a discrepancy between icons and calculator output. If a pupil does not have an adequate mental model or iconic construction of the problem, then attempting the question will be difficult. As a result, he/she may resort to guessing or some other obscure strategy. It is possible that the numbers and symbols located on, and/or generated from, the calculator keypad during an evaluation may cause a distraction and also confuse pupils. Consequently, if a datatree is present, it could further support the iconic structure and overall comprehension – especially an unstable mental model, particularly since there are fewer distracting numbers with a datatree compared to a calculator. Figure 8.11 demonstrates this.

Figure 8.11. Comparison of R.L.'s Calculator and Datatree Representations for Q3

(a) Calculator



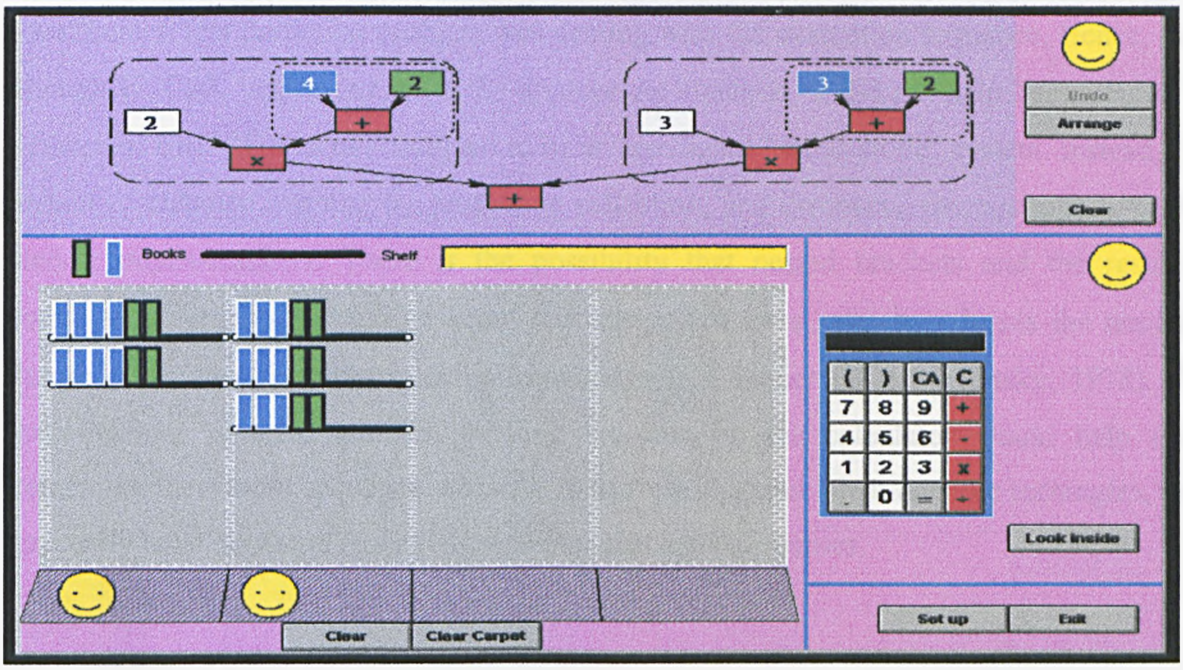
(b) Datatree



R.W. (High Attainment Pupil Group 2) Q3

R.W.'s iconic representation for Q3 (a) was correct, as was her answer of 27 for the total number of books. ENCAL generated the equivalent datatree representation as shown in Figure 8.12.

Figure 8.12. R.W.'s Iconic and Equivalent Datatree Representations for Q3



R.W. interpreted the datatree correctly, because her equation in answer to Q3 (b) was correct – i.e. (2 x (4 + 2)) + (3 x (3 + 2)). However, her subsequent calculation went wrong in the latter stages. She rightly evaluated the inner brackets first - i.e. (4 + 2) = 6 and (3 + 2) = 5, and then accurately positioned the set of outer brackets – i.e. (2 x 6) + (3 x 5). In spite of this, R.W. evaluated 2 x 6 = 12, but then added the 12 to the 3 in the next set of brackets. Consequently, her equation became (12 + 3) x 5 = 15 x 5 = 75. Thus, R.W. completely ignored the second set of outer brackets, indicating that the nested brackets perhaps caused confusion or carelessness. The experimenter commented: "I think you've stuck an extra bracket in there, and there. That's interesting, I'm not just sure how we got that. Well, just for the moment, write that all down for me in there (the experimenter was pointing to look inside the calculator). Write what the answer came to because that's interesting. We'll redo it to check why it did that. Just write the answer down and write those workings."

Since R.W. obtained the correct mathematical expression using the datatree, but did not attain the required answer, it would seem that her mathematical knowledge was insufficient. Her particular problem was dealing with the evaluation sequence. Hence, even though ENCAL helped her write the appropriate equation, it did not help her calculate it. However, ENCAL was not designed to be an intelligent "stand-alone" system. Instead, it is meant to enhance learning by promoting reflection, and discourse amongst pupils-pupils, and/or pupils-teachers. There is the possibility that nested brackets and the resulting expression complexity brought about the calculation error, which endorses the need for tutorial mediation in order to bring knowledge to a higher level (Vygotsky, 1978). The experimenter pointed out that showing brackets on the calculator would help users remember their input sequence, because apart from the equivalent datatree rectangles, they presently have no visual record of calculator bracket key presses.

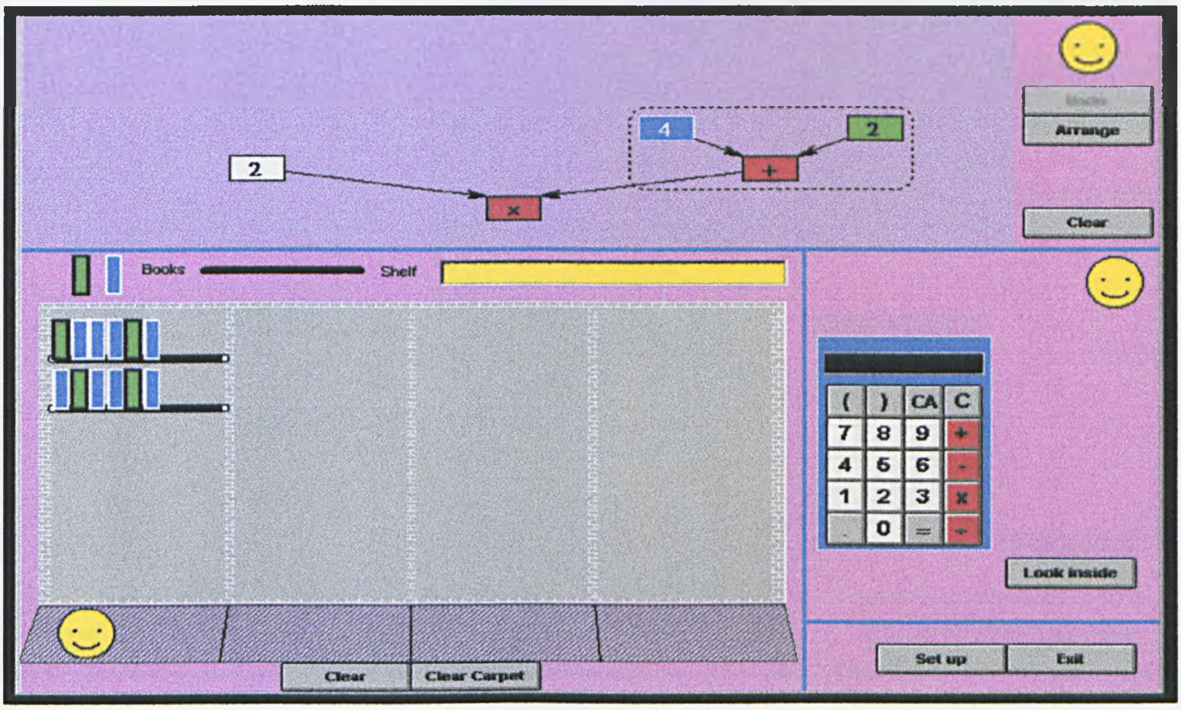
Significance of This Result.

Despite ENCAL enabling a user to obtain the correct mathematical expression, the use of nested brackets impaired the subsequent calculation sequence. In such a case, teacher mediation is essential to further an individual's knowledge. However, ENCAL does support understanding by providing visual representations, the intermediate datatree being crucial. Perhaps showing brackets during calculator calculations would also be helpful.

E.B. (Low Attainment Pupil Group 2) Q3

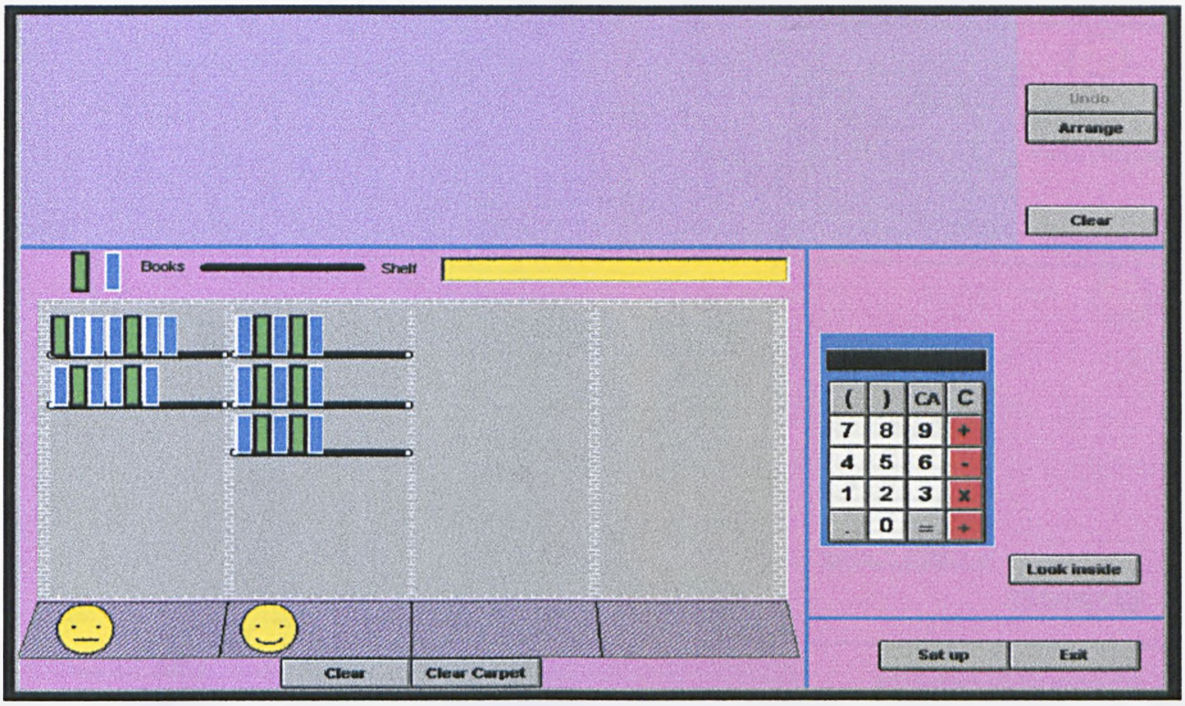
The first bookcase was completed correctly, however E.B. mixed the book colours which may reflect unsystematic thinking. Her random dragging of books is shown in Figure 8.13. Since there was a long pause between finishing the first bookcase and starting the second, the equivalent datatree appeared for the books and shelves present.

Figure 8.13. E.B.'s Initial Iconic and Equivalent Datatree Representations for Q3



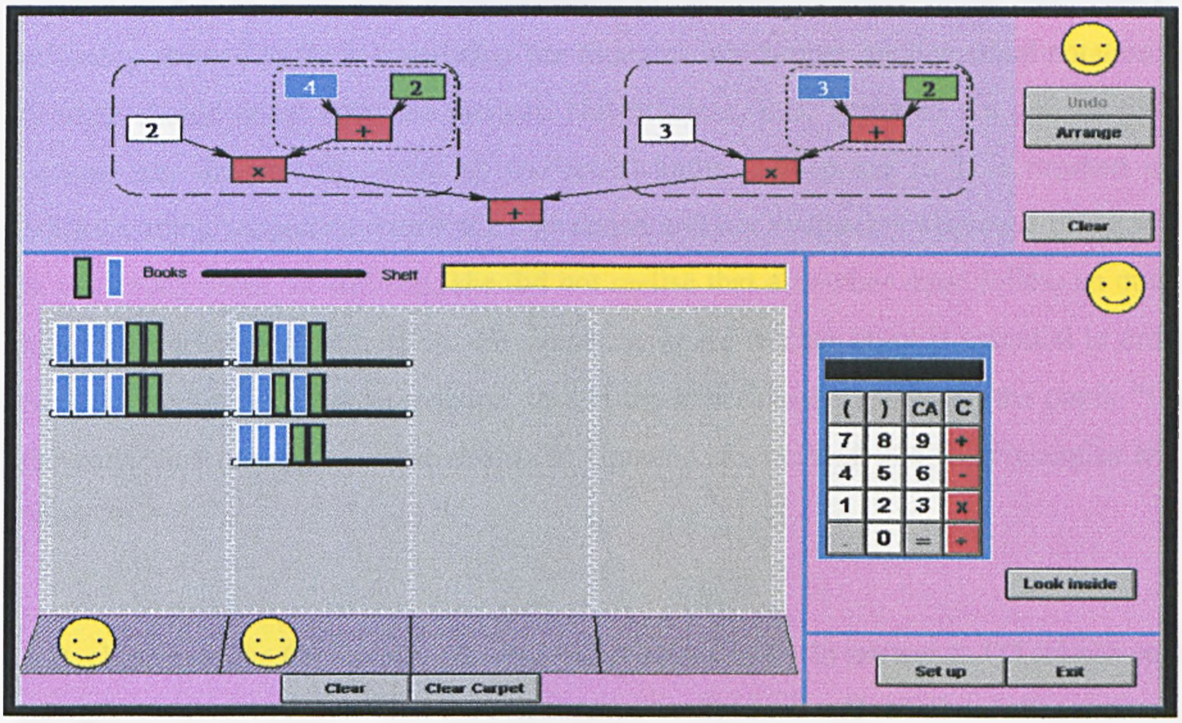
The book colours were also mixed in the next bookcase as shown in Figure 8.14. However, the pattern was regular which tends to reflect more systematic thinking. Also, while completing the second bookcase, E.B. accidentally dropped one blue book on shelf one of the first bookcase. This resulted in an unhappy face because the books were no longer well-formed (i.e. there were more books on shelf one than shelf two). Figure 8.14 demonstrates this.

Figure 8.14. E.B.'s Later Iconic and Equivalent Datatree Representations for Q3



The equivalent datatree did not form due to unequal books in the first bookcase. This is represented by the unhappy face. However, E.B. did not realise this was the reason why the datatree was missing. She then read part (b) of question three and stated: "The datatree is gone...what happened to it?" Although appreciating this, it was not clear whether E.B. understood the relationship between the datatree and forming the mathematical expression. She did not know how to rectify the extra book problem, so the experimenter pointed out that the unwanted book needed to be removed. E.B. then cleared all and started the question again. Alternatively, she could have moved the redundant book to the carpet area and continued. There could be several reasons why she did not do this. Firstly, she did not know. Secondly, she knew, but forgot. Thirdly, she followed the easier option of pressing Clear All (CA). E.B.'s second attempt at question three is shown in Figure 8.15.

Figure 8.15. E.B.'s Second Iconic and Equivalent Datatree Representations for Q3



E.B. initially wrote down the answer of 27 in answer to Q3 (a) which is correct, but then she recounted and changed her response to 26. Her iconic display is correct, however the books colours on the first two shelves of the second bookcase are haphazardly positioned.

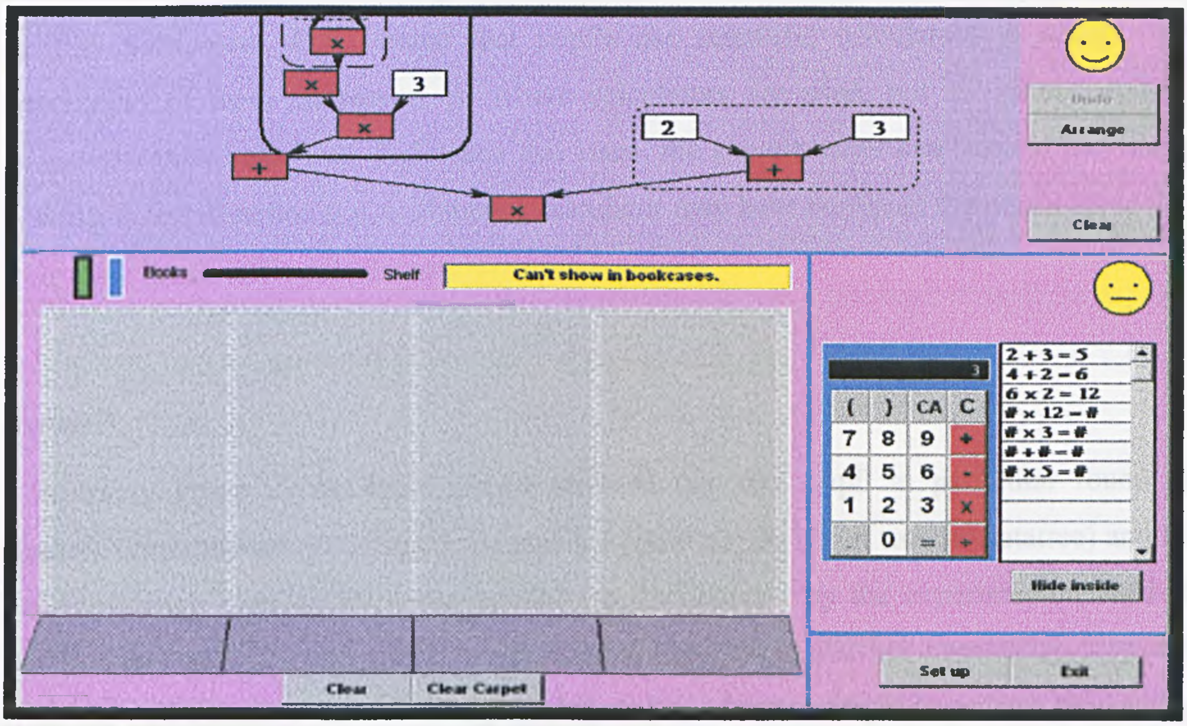
Despite having the correct equivalent datatree, E.B.'s expression for Q3 (b) was:

(4 + 2) (2 x) + (3 x) (2 + 3. The experimenter pointed out that E.B. said she could not do it, and so she just wrote the expression in the same way as she would work it out. This suggests that E.B. did not notice the significance of the datatree, and therefore teacher input at this juncture would be needed. Predictably, she did not have the correct number of brackets for the required expression and she also missed the closing bracket from her final step. Having said this, the expression reveals the logic behind her thinking as follows. The 4 + 2 represents the four blue books and the two green books. The 2 x indicates that 4 + 2 has to be multiplied by 2 since there are two sets of four blue books and two green books.

Similarly, E.B. realises that there are three sets (i.e. 3 x) of two green books and three blue books (i.e. 2 + 3). Since E.B.'s interpretation of the problem requirement was correct, her difficulty seemed to lie in translating her thoughts into proper mathematical notation. The datatree should assist with this process. Nevertheless, E.B. appeared not use the datatree because she did not: (a) notice it; and/or (b) know how to use it. This resulted in her brackets being misplaced within the expression and her evaluation sequence being written in the wrong order. In addition, she did not realise that a number must follow an operator before a bracket is closed. These are situations where teacher input is essential in order to further learning and thus knowledge. In fact the experimenter reminded the pupil: "Using the connected rectangles, write down the equation. Remember that the rectangles are the brackets."

E.B. clicked on the CA button, and thus cleared the screen. In answer to Q3 (c) she entered the data as given in Q3 (b) into the calculator: (4 + 2) (2 x) + (3 x) (2 + 3. She omitted the final bracket. The resulting ENCAL screen is shown in Figure 8.16. The experimenter urged E.B. to use Look inside the calculator, and write down her sequence of key presses. E.B. entered the mathematical expression exactly the same as she had written it down in answer to Q3 (b). This is the reason why: (a) the calculator Look inside display was erroneous (e.g. the # symbols); and (b) the iconic representation did not appear.

Figure 8.16. E.B.'s Calculator Input and Equivalent Datatree Representation for Q3



Significance of This Result.

The datatree does not help with the formulation of an expression where prior mathematical knowledge of this topic is weak. Mediated learning is therefore necessary (i.e. teacher intervention), which highlights the fact that ENCAL is a pupil-teacher system.

8.4 Experimenter's Feedback

Group 1

The experimenter pointed out that the wording of question three was problematic. Consequently, difficulty with understanding was compounded where participants did not read very well, as with pupil L.A. It was also noted down by the experimenter, that whilst R.L. answered the second demonstration question, he clicked on the calculator addition key instead of the multiplication key which resulted in a variety of incorrect answers. However, when the experimenter mentioned brackets, R.L. achieved the correct answer. Yet in test question two, R.L. used the brackets in an incorrect order. He tried again but still got it wrong, and then said to the experimenter: at school we only use calculators later on. Another participant, D.P., stated that pupils use scientific calculators in school. The implication of these statements is: where calculators are used, not all school students understand their behaviour. Reference this issue, the experimenter emphasized that use of ENCAL's four-function (i.e. arithmetic) calculator may have confused the pupils since they were used to scientific calculators.

Group 2

Concerning group 2, the experimenter stressed that one pupil in particular found the datatree representation daunting. E.B. said she could not do it (i.e. use the datatree) and lost her confidence. Conversely, other pupils – as documented by the experimenter, used the datatree and obtained the correct mathematical expression. Further interesting comments from the experimenter are: the changing shape of the datatree might have been misleading; the slowness of the software may have affected students' thought processes; the connecting arrows were probably confusing; and the momentary disappearance of rectangles during datatree construction could be a problem.

The experimenter's comments regarding the datatree in particular, highlight areas of the ENCAL design which require further attention. A way of overcoming the changing shape of the datatree is to keep the existing structure intact and then form a new tree independently. This would also prevent the perplexing temporary removal of rectangles from the screen. The connecting arrows could be confusing to pupils, and a way of alleviating this difficulty would be to have a teacher explain their significance.

8.5 Summary/Conclusion

The results in Chapter 7 and the analysis of errors in this chapter suggest that two important factors are beneficial to learning when carrying out mathematics involving order of operations and brackets: (a) the datatree; and (b) a more complicated but informative (i.e. educational) interface which arises because of the datatree – that is, a more enlightening learning structure which is comprised of "boxes" for symbols and connecting arrows. Superior accomplishment when using the datatree was evident due to the fewer operator and brackets errors. Specifically, it was found that neither low nor high attainment pupils in group 2 (datatree) had operator errors, whereas pupils in group 1 (no datatree) did have these errors. In addition, only two group 2 pupils had brackets errors, but five group 1 pupils had such errors. These results suggest that ENCAL has a very beneficial effect on performance. This seems to be due to the addition of the linked datatree representation. Exactly how this additional feature might be of assistance to the learners is one of the topics for discussion in the next chapter.

Additionally, we can ask whether the participants have learned anything. Since the study is not a longitudinal one we cannot demonstrate retention of information, nor can we show improved performance on a post test. However we have argued already that this would take us into territory that lies outside the main focus of the work. What we can say is that we have an outcome that may well be relevant to effective learning that can be pursued in later work. This again will be raised in a later chapter.

Some design issues which need further consideration were also identified during the analysis of errors. These issues were associated with the datatree, and with the nature of MELR's themselves. These points are discussed in the next chapter.

Chapter 9

Discussion

9.1 Introduction

Greeno (1991) points out that "students who adapt successfully to the requirements of the mathematics classroom learn to listen, watch, and mimic effectively. If we believe that these receptive forms of activity are insufficient as outcomes of mathematics education, we may be forced to consider fundamental changes in the kinds of activity that students engage in as they learn mathematics." (P. 76). The ENCAL software put forward in this thesis has been designed not only to help overcome such limiting learning activities, but to encourage enquiry. In Chapter 8 the participants' analysis of errors regarding the final evaluation were considered. With these in mind, this chapter provides a general discussion of the effectiveness and shortcomings of ENCAL. Initially, a summary of the original design ideas is given. This is followed by sections on what ENCAL does deliver and what ENCAL does not deliver. Some implications for other interactive systems are discussed next. Then consideration is given to how the current version of ENCAL is an improvement on earlier representations. Learning and translation issues using ENCAL are discussed next. Finally, the chapter is summarised and conclusions are given.

9.2 Summary of Original Design Ideas

Firstly, the *underlying design aim* of ENCAL is to influence individuals' construction of mental schema and thus mathematical thinking, by means of three computer-based representational styles. Refer to Chapter 4 for a complete description of the design aim.

Secondly, the system is designed to promote learning within a social context, as opposed to an individual context, by offering students: meaningful interaction – including teacher interaction, peer interaction, and interaction with the system itself; active thinking; reflective thinking; and partial interface feedback (Chapter 4 gives further details).

Thirdly, ENCAL was designed around multiple, equivalent, linked representations (MELRs). This is because they are particularly suitable for learning and understanding concepts and procedures in the domain of arithmetic, as outlined in Chapter 2.

Fourthly, the three representational styles designed for ENCAL may be used in any sequence. However, as stated in Chapter 5, in order to enhance understanding it is assumed that users' preferred movement between the three representations is from concrete to abstract as follows:

1. *icons* (concrete) to dataflow;
2. *dataflow* (intermediate structure) to calculator;
3. *calculator* (abstract) back to dataflow.

9.3 What Does ENCAL Deliver?

The underlying design aim given in Section 9.1 is an implicit (i.e. hidden/internal mental/cognitive) process. However, the results of the final evaluation demonstrate explicitly (i.e. openly) that the use of all three computer-based representations influenced group 2 participants' thinking. Therefore, conceivably over time and with teacher intervention, the three representations will facilitate the growth of appropriate mental schema with this type of mathematics. The three representations in the ENCAL system are designed to be open-ended (i.e. exploratory) and not wholly explicit. This provides the opportunity to promote active thinking, and facilitates learning via reflective abstraction and social interaction.

Since the evaluation questions were answered by participants without teacher support or peer interaction, learning within a social context was not apparent. However, it was evident from the pupil and experimenter discourse (as highlighted on the video analysis) that questions were often asked or statements made by students in order to seek clarification. This indicates that interaction with ENCAL resulted in an open-ended (i.e. unconstrained) outcome, which offers support to the second design idea of users being active and reflective.

The better results of group 2 (who had a datatree) compared to group 1 (who had no datatree) demonstrate the pedagogical effectiveness of the use of all three representations, especially the intermediate datatree. This was particularly apparent with answers to question 3 (b), where the datatree proved to be a necessary source of information to achieving the required expression. Such a result lends support to the notion that MELRs facilitate the understanding of mathematical concepts and procedures, particularly where users move through the representations from concrete to abstract (see Section 5.2). It was noted however that one or two children went into a state of panic when confronted with the datatree. This shows that not only did these children have trouble with the problem, but the datatree representation created anxiety.

9.4 What Does ENCAL not Deliver?

The final evaluation highlighted the need to keep previous states of users' ENCAL interactions. This is because users were not given the opportunity to retain an existing representation once it had been changed, and it was felt that this would have provided a further means of assisting learning. That is, currently, the history of an initial representation disappears as soon as any modification takes place, so comparison between the two states (i.e. old and new) is not easy since the new view diverts a student's attention and thus his or her short-term memory. The point of MELRs is that students do different kinds of reasoning in each of the three representational states in order to help with understanding. ENCAL was designed to encourage users to move (i.e. translate) between the three representations from iconic to datatree to calculator (i.e. concrete to abstract). In order to maintain equivalence throughout when working is later modified, earlier states change. Unfortunately, it is then impossible for individuals to revisit their thinking by reviewing older versions of the representations. So the problem is how to maintain enough history of earlier states so that they can be revisited while: (a) keeping the linked property; and (b) not overwhelming users with clutter.

Also, when users handle nested brackets with ENCAL, teacher support was found to be crucial. This is due to the system being unintelligent and to some extent unrestricted in terms of user actions. In other words, ENCAL on its own was not designed to be a tutor for handling nested brackets – particularly, the positioning of brackets in an equation and the subsequent calculation sequence. Although the current state of ENCAL is in accordance with the original design ideas (in particular, being open-ended to stimulate thinking, and the need for pupil-teacher interaction), some users found nested brackets confusing. Therefore, in order to avoid users being in a helpless situation, perhaps ENCAL should address this issue and offer some guidance, for example only if sought by users. However, it must be emphasised that ENCAL is not designed to simplify the learning process in some way, or to weaken the role of the teacher.

Even though datatree construction was not included in the final evaluation, the pilot test showed that this activity was difficult for users, since ENCAL does not offer support for this task. As with nested brackets, ENCAL should ideally provide early help, but much hands-on experience would be required in order for users to become proficient and thus gain maximum benefit. However, in line with the above original design idea, *meaningful interaction* through teacher support will ultimately be needed, particularly in the early stages of building datatrees.

Following the pilot test, it was decided that a datatree and its equivalent book and shelf icons would not alter their structure when switching between the left-to-right and the BODMAS calculators following datatree construction. This was done in order to avoid confusion. The calculator logic selected by a user (e.g. BODMAS) is therefore not re-represented by the datatree or the icons. Instead, the original calculator logic (i.e. left-to-right) remains in the initial left-to-right state. However, when the equals key on the calculator is clicked, the answer displayed changes accordingly. On reflection, if the calculator answer changes due to a different logic, the icons and datatree should change also in order to maintain equivalence between the representations. Nevertheless, such a move will be feasible only if the original representations can be maintained. Therefore, two ongoing problems with ENCAL which require attention are: (a) updating iconic and datatree structures so that they conform to the current calculator logic; and (b) preserving the previous iconic, datatree, and calculator states for comparison purposes.

9.5 Some Implications for Other Interactive Learning Systems

Based on the findings from ENCAL, suggestions for other interactive learning environments are put forward in this section.

Firstly, all representations at the interface should be *editable*. The reasons for editing are to: (a) support understanding; (b) ensure that mistakes made during user input (which for ENCAL involved iconic and datatree construction and calculator data entry), are easy to change so as to assist learning and avoid user frustration; and (c) make building difficult structures (such as datatrees in ENCAL) achievable.

Secondly, the intermediate datatree was designed to make the cognitive process of translating between different representations easier, but not too easy, so that a little thinking is required by a user. Consequently, the datatree is a necessary but complicated representation. It therefore has a set *order of allowed actions* with the operators and entities to facilitate construction. A similar procedure could be adopted with other complex interactive environments. Nevertheless, tree building is still quite difficult, and therefore teacher support would be required until a user becomes proficient.

Thirdly, the structure and functionality of multiple representations should preferably be *equivalent* and *linked* so that comparisons between concrete and abstract forms are readily observable in order to assist learning.

9.6 How the Current ENCAL Improves on Previous Representations

Some of the problems with earlier ENCAL representations are listed below. The representations did not work from educational or usability perspectives due to the reasons given. ENCAL was subsequently amended to the current form.

Iconic

In order to avoid confusion, a *textual prompt* was needed in the iconic representation to state that the number of books and/or shelves selected by an individual was too great to be shown.

Shelf icons could not be *dragged and dropped* like the book icons. This meant that there was a disparity between different icons and their movement.

It was thought that with 5+2 (for example), there should be no need to have 5 books in one bookcase, and 2 books in another bookcase to represent addition. Instead, it was decided that the sets of 5 and 2 could be represented with *different colours* in one bookcase. Consequently, there could be 5 blue books and 2 green books on one shelf in one bookcase. In this case, zero would be represented by omitting books alongside other books on a shelf - so 5+0 is represented by 5 blue books, and no green books are shown.

Dataflow

Parentheses (i.e. rectangles – "boxes") could not be inserted around specific numbers and operators in the dataflow representation after it had been constructed, such as with: (4+2)x3+(2+1)x2.

Users did not have the freedom to be able to *clear boxes* during tree construction in the dataflow representation. The UNDO key did not allow users to delete a box once it had been inserted, instead the whole tree would clear.

To overcome these problems, a datatree *editing* system was incorporated, where a user should at least be able to:

- move a "box" containing a number or operator to a new location;
- delete a connecting arrow;
- insert a new connecting arrow from a number or operator "box" to an operator or number "box" respectively.
- put numbers or operators into "boxes" before having to connect arrows between the "boxes".

Calculator

The *record of calculation steps* was permanently shown at the side of the calculator. However, it was decided to incorporate a button so as to hide or show this if desired.

All Three Representations

A *tick and a cross* were changed to *happy and sad faces* to signify a user's well-formed or not well-formed representations. It was felt that a tick and a cross could be mistaken for a correct or an incorrect answer, as opposed to representations being well-formed or not. However, it was noted during the evaluation sessions that pupils tended not to refer to these prompts, and so they seemed to be of little help. Having said this, with further experience of ENCAL, it is assumed that users would rely more on their presence.

9.7 Translation and Learning Issues Using ENCAL

"One of the main problems in any learning is the need to 'translate' concepts from a symbol system, such as language or numbers or graphical representations/models into meaningful mental schema" (Somekh, 1996, p.12). ENCAL has been designed to facilitate translation (i.e. re-representation) from textual word problems to iconic, datatree, and calculator graphical representations and thus overcome the translation problems highlighted by Lesh, *et al.* (1987) – see Chapter 2. Information from the three external representations is then translated into meaningful internal mental schema helped by the computer-based learning environment (this combined process is referred to as *distributed cognition* since cognitive tasks are distributed across both the internal and the external representations). This translation process is helped due to the learning being *situated*. In other words, ENCAL provides a microworld or simulation within which particular mathematical experiences (i.e. order of operations and use of parentheses) take place. In order to achieve translation, the rationale behind learning using ENCAL is discussed below.

ENCAL provides three external representations comprised of various symbols to facilitate the transfer of learning to internal/cognitive representations. However, user translation between the different external representations is a complicated cognitive process. This is pointed out by Van Labeke and Ainsworth (2001), but they state that even young children can benefit (i.e. learn) from multiple representations provided that they are given suitable illustrations. Van Labeke's and Ainsworth's quest it is to learn and identify more about the actions of students (in particular, the translation issue) from a cognitive perspective, whereas here this aspect is considered with education as the main focus. The design of representations in their study and in ENCAL tend to reflect these differing approaches, as explained below.

The external representations in ENCAL are based on the philosophy of constructivism (refer to Chapter 1 for further information) in order to assist and maximise both the learning process and the development of internal representations. In other words, users have the opportunity to become *actively* involved in building (i.e. *constructing*) concrete icons, intermediate datatrees, or abstract mathematical expressions. It is this active, constructive process which supports: (a) cognitive development during the learning process; (b) translation between linked but structurally different external representations; and (c) reading-off information. In contrast, the external representations put forward by Van Labeke and Ainsworth (2001) are not actively constructed by users. Instead, students input values and parameters (e.g. to forecast predator and prey relationships up to a particular year), and consequently three external representations (a graph, a table, and a phase-plot) are updated. An individual then reads-off the changes and considers the connection between the representations.

Despite the efficacy of such multiple representational systems, Cox (1999) states that the "process of constructing and interacting with an external representation is a crucial component of learning" (p. 347). He points out that the construction of external representations can assist problem solving by: (a) re-ordering the information in ways useful for solutions; (b) laying out the range of possible uses of the information; (c) making missing information explicit; (d) representing implicit information explicitly; and (e) facilitating translation between representations. However, he also indicates that external representations must be appropriate to the requirements of a given task, particularly to facilitate *read-off.* In addition, Reisberg (1987) refers to the process of constructing an external representation as being a procedure for enlarging a person's understanding. Grossen and Carnine (1990) also state that the process of constructing a representation helps transform students' understanding of a problem. The ENCAL system endorses this statement. For example, ENCAL allows users to construct concrete icons in order to represent familiar real-world objects. The icons are a means of re-representing textual information. A student then progresses to constructing a datatree which re-represents the iconic information. Finally, the formal calculator expression is input which re-represents the intermediate datatree. Related to this issue are the concepts of *modality* and *capturing abstraction.* According to Cox, Stenning & Oberlander, (1995), the ability of an external representation to depict abstract information contained within analytical word problems is more important than the modality (i.e. type) of an external representation. However within ENCAL, both a representation's ability to depict abstract information and its modality are significant factors. This is because word problems are introduced first, followed by the three external representations - icons, datatree, and mathematical calculator notation (modality). A representation's modality is crucial to ENCAL's environment since it is the modality which helps guide users through the subsequent learning process from concrete to abstract (ability to convey abstract information). Nevertheless, the expressiveness of the iconic modality is limited. In addition, datatree representations do not give products of operations, whereas either of the calculators do. This was due to iterative design decisions, having taken learnability issues into account. Consequently, it may be seen that ENCAL's three external representations are not exactly informationally equivalent.

The fewer hidden dependencies (refer to Chapter 2) in ENCAL reinforce the constructivist approach. For example, a user builds the graphics so that parentheses are represented as bookcases (concrete), dotted rectangles (intermediate), or algebraic brackets (abstract). Thus, a calculation *sequence* does not have to be inferred since this is dependent upon the information present. However, the three representations in the read-off simulation described above are not user constructed, and as such will be less familiar. Consequently, a certain amount of user inference will inevitably take place. In this case, dependencies both within and between the representations tend to be hidden, and thus have to be inferred.

With a read-off only interface, it may be assumed that users are passive since they observe as opposed to construct. However, Brna, Cox, and Good (2001), point out that observation is not inevitably passive, since a person may interpret a diagram without having to physically manipulate it. Such interpretation may be difficult though if the representation is not constructed. This seems to be particularly important in a learning environment. For example when using ENCAL, entering + on the calculator results in + appearing on the datatree. This constructive activity gives a user the expectation that as well as addition, another number is required, and therefore indicates that + has a dual function.

Although constructivism may be beneficial to learning, Brna, *et al.* (2001) state that there is a tension between learning unfamiliar diagram systems (i.e. representations) and learning a *new* educational topic, because students are concurrently formulating mental models of the domain knowledge and the representations. In this thesis students had knowledge of the mathematical subject (i.e. it was not new), yet they may have had incomplete conceptions or misconceptions. Nevertheless, the results of the experiment (see Chapters 7 and 8) showed that interacting with unknown representations created some confusion, but translation between them was evident due to the beneficial results of the datatree group. Despite these results supporting learning, Lowe (1994) points out that specific representation systems do not help students, but make the task of understanding much harder since he or she is struggling with both learning knowledge and the representation(s). Ainsworth, *et al.* (1996) endorsed this finding by demonstrating that providing multiple representations does not necessarily guarantee learning success.

However, the explicit provision of domain knowledge in ENCAL helps both the translation and the learning processes. For example, in the iconic representation different book sets are represented using different colours, and multiplicand and multiplier are distinguished using books and shelves. With the datatree, the operator and number "boxes" are connected by arrows in order to highlight the direction of a calculation, and dotted rectangles represent parentheses. Finally, the calculator has the operator keys marked in red in order to distinguish them from the other keys. It must be remembered that the provision of such domain knowledge has been designed not to be overly explicit. This is because the environment is purposely open-ended to achieve the desired learning process of encouraging reflection, discussion, and question-asking.

The results of an experiment by Cox and Brna (1995) also found that multiple external representations are effective, despite students' reasoning behind external representation selection being heterogeneous (i.e. varied). In fact, Schnotz, Picard, and Hron (1993), argues that diagrammatic representations are fundamental to the development of appropriate mental models. He found that unsuccessful students tended to use graphic information to a lesser extent, whilst successful students constructed mental models using the diagrammatic representations. The results of the ENCAL experiment tended to reflect this position. That is, high attainment pupils successfully read datatree information (in particular, question three) which implies that they built appropriate schemata of the situation. In other words, pupils translated the iconic diagram into an appropriate datatree representation, and then formulated the correct mathematical expression. This demonstrates that actions were translated between the three connected external representations, and also shows that mathematical thinking was assisted by non-verbal imagery (Goldin, 1992b).

Cox (in Brna 2001) believes that graphical representations compared to textual information helps *reflective* students consider more than one model of the information present, although he does state that it is still unclear as to the way in which diagrams play a part in this process. It is possible however that interactive software such as ENCAL provides mental scaffolding due to the representations, which gives students the encouragement to reflect. On this issue, Somekh (1996) states that at the lowest level, such scaffolding provides the motivation to work on tasks for longer periods. Furthermore, it must not be forgotten that ENCAL provides a social learning situation. This is because use is made of *cognitive tools* (e.g. text, language, the three representations, and teacher/pupil interaction). According to Vygotsky (1986), learners have limits to their current level of knowledge (this is known as the "zone of proximal development", p.187), and they are unable to proceed beyond this without teacher or peer intervention. The idea being that the next time the person comes across a similar learning situation, he or she will be able to proceed without such social scaffolding.

9.8 Summary

The ENCAL design promotes learning via three external computer-based representations. Distributed cognition is fundamental to the underlying rationale of the system since the multiple, equivalent, linked external representations (i.e. iconic - concrete, datatree - intermediate, and calculator - abstract) have been designed to influence individuals' internal representations (i.e. mental/cognitive processes). The results of the final evaluation showed that the *intermediate* datatree was particularly beneficial to participants' performance. However, ENCAL does not keep previous representational states for reference once a representation has been modified by a user. In addition, the ENCAL unintelligent system does not offer assistance with nested brackets, and it was evident that in such circumstances teacher support is essential. It was also found that teacher help will be needed for early attempts at datatree construction. It is recommended that: (a) all representations should be editable; (b) complex constructions have a set order of allowable actions; and (c) multiple representations are equivalent and linked. Various iterative improvements were made to the ENCAL interface throughout the design process.

ENCAL enables users to translate (i.e. re-represent) information: (a) between the three external representations; and (b) from the external representations into meaningful mental schema. The ENCAL microworld has been designed to assist this process and thus maximise learning based on the thinking behind constructivism. Although the unknown representations caused some confusion, translation between them was evident particularly with those participants provided with the intermediate datatree. In addition, the explicit provision of domain knowledge in the three representations facilitated the translation process. The ENCAL computer-based interactive environment also affords both mental scaffolding which encourages reflection, and a social learning environment through the use of cognitive tools (i.e. text, language, the external representations, teacher–pupil, and pupil–pupil interaction).

9.9 Conclusions

The use of all three graphical multiple equivalent linked representations (i.e. icons, datatree, and calculator) have beneficial effects on participants' mathematical performance. The intermediate datatree serves as a crucial connection between the concrete icons and the abstract calculator notation. The unintelligent (i.e. open-ended) behaviour of ENCAL promotes learning by means of reflective abstraction and social interaction. Constructivism is central to the learning process which is provided through ENCAL, since activity (i.e. user construction of iconic and datatree diagrams) appears to be an important factor in understanding, and thus the formation of mental models. In general, the ENCAL computer-based learning environment offers an effective link - enabling translation of information, between its three external representations and subsequent knowledge which develops in the form of individuals' schemata.

The next chapter considers future work concerning ENCAL, and the contributions made regarding computer-based learning research.

Chapter 10
Contributions and Future Work

10.1 Introduction

The following chapter presents contributions made during this research, and the possible future work which may be carried out to further improve the ENCAL computer-based learning system. Initially, the contributions are stated specifically with regard to design. Future work is then considered concerning constructivism issues. Finally, further suggestions associated with future work are put forward.

10.2 Contributions

(a) The Design and effectiveness of the Multiple Equivalent Linked Representations (MELRs)

The result of the study has shown that it is possible to design a successful system (i.e. ENCAL) based on a combination of educational and cognitive theories. The educational theories included: constructivism, meaningful interaction, reflective abstraction, and pupil-teacher/pupil-pupil discourse. The cognitive theories were based on: (i) external representations influencing internal schemata – i.e. the translation of information between different external representations was assumed to influence individuals' mental models, where concrete to abstract translation and thus understanding during learning was facilitated using three external graphical diagrams (icons, datatree, calculator); and (ii) the cognitive dimensions method of evaluation which was used for the interface design.

Individuals had a choice of graphical constructs. Firstly, the icons enabled word problems to be represented diagrammatically. However, the book and shelf icons behaved differently at the interface. In particular, the method of removing shelves and books from bookcases completely differed. Such inconsistencies were problematic and caused confusion to some users. Nevertheless, this type of problem has alerted the need to ensure *consistency* throughout icon use. Secondly, the datatree – in particular the "boxes" and the dotted rectangles, were new to users. Also, the mapping of recursive datatree constructs (expressed as parentheses in calculator notation) onto graphical representations (e.g. "boxes") was only partially successful. Consequently, future work concerning datatree design and its use is needed.

Despite these reservations, the research provides a significant contribution to the growing number of studies on the use of multiple linked external representations in educational contexts. In particular, this study shows an effect which has potentially significant implications for further research in the value of multiple representations.

(b) The Use of Calculators

A significant contribution has also been made to the ways in which the teaching of number can make use of calculators. Specifically, the ENCAL system provides an original way of working with calculators, which has the potential for helping children learn about both number and how calculators work.

(c) Cognitive Dimensions

ENCAL was designed with the help of cognitive dimensions, which should be of value to those seeking to develop the notion of cognitive dimensions further, especially in an educational context.

10.3 Future Work

Although ENCAL is currently a viable computer based learning environment, further modifications to the system would enhance learnability. In addition to the ENCAL problems outlined in Section 9.4, two areas of future work based on the above contributions are described below. Further suggestions are then given concerning possible future work.

Constructivism

The reasoning behind the constructivist design of ENCAL is to gradually *fade the scaffolding* provided by the icon, datatree, and software calculator constructs. Thus, the aim is for individuals to interact with a normal hand-held calculator over time and understand both the mathematical input and the calculator behaviour. A long-term study would therefore be appropriate in order to investigate whether the graphical representations could indeed be faded out.

Icons and Datatrees

Previous research indicates that the construction of computer graphics (which in ENCAL are iconic and datatree representations) by a user better facilitates the translation (i.e. mapping) of symbolic information to the formation of mental schemata and thus learning.

In view of this, the book and shelf icons need to be redesigned so that they allow similar user actions during construction and editing. In addition, future work should be involved with: (a) developing a less complicated yet editable means of constructing datatrees; and (b) ensuring that the construction of icons and/or datatrees form an integral part of the interactive process. This pedagogy will help reinforce the structuring of knowledge, and would be the preferred learning process, as opposed to having students read off information from automatically constructed graphics.

ENCAL should also be modified to include – and ÷ as well as the existing + and x. This would only require an extended concrete iconic representation, e.g. sharing out a pile of books, since the underlying mechanism is already able to handle the arithmetic of all four operators via the datatree and the calculator.

10.4 Further Suggestions

Question Text

In order to facilitate understanding, ENCAL could update question text to match user actions when interacting with the iconic, dataflow, and calculator representations. However, the problem then would be that the original question text would have to be stored somewhere so that it could be referred to by users.

The Use of Representations

The use of all three representations concurrently may be confusing. In such a circumstance, it may be useful if any one representation is temporarily negated. A user could then choose which two representations to interact with.

Calculator Modes

The two calculator modes (i.e arithmetic, and scientific) could be made accessible in the calculator representation. Currently, a user has to return to the set up menu to change calculator logic, and consequently all current information is lost and so the learning process is interrupted. An operator hierarchy in a tree format could be shown on the screen concurrently for each calculator logic. Thus, a user would be able to see at a glance the sequence the calculator will use to evaluate an expression. Whichever calculator logic is chosen will be reflected in the other two representations, thus maintaining equivalence.

Operators

Currently only addition and multiplication are used in the ENCAL design. The development of subtraction and division would be a next step for upgrading the design to include all four arithmetic operations. However, further icons would have to be incorporated into the iconic representation to make the system versatile enough to cope with this situation.

The Provision of Additional Teacher Material

In order to gain maximum benefit from ENCAL, students will require frequent teacher support, e.g. with the construction of datatrees. Therefore, it is recommended that teachers should be provided with a workbook or a resource-pack so that they are able to make the best use of ENCAL in class.

Hypothesis Testing

Hypotheses testing could also be carried out. For example, the use of a four-function (i.e. arithmetic) calculator could be compared with the use of multiple, equivalent, linked representations to solve particular problems. A relevant hypothesis could be as follows.

> ***The use of a four-function (arithmetic) calculator causes confusion when evaluating arithmetic expressions, compared to the use of multiple, equivalent, linked representations of the ENCAL computer-based learning system.***

The independent variable would be the interface style. The dependent variable would be the number of errors made in terms of incorrect final answers to calculations. The experimental group would have only the calculator representation to work with (i.e. the interface would be manipulated to show only the calculator), whilst the control group would not have this manipulation, but would have access to the complete ENCAL system. Thus, it would be the interface manipulation which is responsible for any differences found between the groups.

In addition, the following hypothesis may be tested in order to confirm the assumption of Hiebert and Lefevre (1986), that procedures (i.e. calculation sequences and order of operations) may or may not be learned with meaning.

The evaluation of an arithmetic expression may be successfully performed even before achieving a full interpretation of the meaning of an expression.

Comparative hypotheses testing may also be carried out. For example, the usefulness/effectiveness of one or more of the three representations could be compared with each other. Also, the usefulness of one or more representations of the system could be compared to traditional calculator and paper and pencil methods.

History

It is considered essential that the history of a representation is maintained for future reference once any modification has taken place (Section 9.4 gives further details). If this occurs, then constructed iconic and datatree representations may be updated following a change of calculator logic. The representation of history (preferably in editable form) is being explored within the Human Computer Interaction community, but has made little headway within Computer Based Learning. Such work could also include the use of *role back execution.* This could be incorporated into the calculator representation so that a user has the opportunity to return and edit his or her input.

10.5 Summary

The two fundamental contributions were: (a) the design of ENCAL which was based on an amalgamation of educational and cognitive theories; and (b) the helpful and constructive use of multiple equivalent linked representations. Future work should include: (a) a long-term study to assess the effectiveness of fading the constructivist scaffolding (i.e. the three computer-based representations) over time; (b) the redesign of the construction and editing of icons and datatrees. Further suggestions for future work are stated, and specifically should focus on the need to develop a means of maintaining the history of work previously carried out but subsequently changed.

References

Anderson, J. R. (1993). *Rules of the Mind*, Erlbaum, Hillsdale, NJ.

Agate, S. J., and Drury, C. G. (1980), Electronic calculators: which notation is better? *Applied Ergonomics*, 11, 1, 2 - 6.

Ainsa, T. (1999). Success of Using Technology and Manipulatives to Introduce Numerical Problem Solving Skills in Monolingual/Bilingual Early Childhood Classrooms, *Journal of Computers in Mathematics and Science Teaching*, 18, 4, 361-369.

Ainsworth, S., Wood, D., and Bibby, P. (1996). Co-ordinating Multiple Representations in Computer Based Learning Environments. In: Brna, P., Paiva, A., and Self, J. (Eds.), European Conference on Artificial Intelligence in Education, pp. 336-342.

Attisha, M. and Yazdani, M. (1983). A Micro-Computer Based Tutor for Teaching Arithmetic Skills, *Instructional Science*, 12, 333-342.

Attisha, M. and Yazdani, M. (1984). An Expert System for Diagnosing Children's Multiplication Errors, *Instructional Science*, 13, 79-92.

Bell, A. W., Costello, J., and Küchemann, D. (1983). *Research on Learning and Teaching: Part A: A Review of Research in Mathematical Education*, NFER-Nelson.

Benbasat, I., and Todd, P. (1993). An experimental investigation of interface design alternatives: icon vs. text and direct manipulation vs. menus, *International Journal of Man-Machine Studies*, 38, 3, 369-402.

Boohan, R. (1994). Interpreting the World with Numbers: An Introduction to Quantitative Modelling. In: Mellar, H., Bliss, J., Boohan, R., Ogborn, J., and Tompsett, C. (Eds.), *Learning with Artificial Worlds: Computer Based Modelling in the Curriculum*, The Falmer Press, London, Washington D. C.

Boohan, R., and Brosnan, T. (1994). Reasoning with a Quantitative Modelling Tool. In: Mellar, H., Bliss, J., Boohan, R., Ogborn, J., and Tompsett, C. (Eds.), *Learning with Artificial Worlds: Computer Based Modelling in the Curriculum*, The Falmer Press, London, Washington D. C.

Brna, P., Cox., R. & Good, J. (2001). Learning to think and communicate with diagrams: 14 questions to consider. *Artificial Intelligence Review, Vol. 15*, 115-134.

Brown, J. S., and Burton, R. R. (1978). Diagnostic Models for Procedural Bugs in Basic Mathematical Skills, *Cognitive Science*, 2, 2, 155-192.

Burton, R. R. (1982). In: Yazdani, M. (1987). Artificial Intelligence, Powerful Ideas and Children's Learning. In: Rutkowska, J. C. and Crook, C. (Eds.), *Computers, Cognition and Development: Issues for Psychology and Education*, John Wiley and Sons, Chichester, New York, Brisbane, Toronto, Singapore.

Burton, R. R., and Brown, J. S. (1982). An investigation of computer coaching for informal learning activities. In: Sleeman, D., and Brown, J. S. (Eds.), *Intelligent Tutoring Systems*, Academic Press, London, New York, Paris, San Diego, San Francisco, Sao Paulo, Sydney, Tokyo, Toronto.

Carroll, J. M., and Mack, R. L. (1985). Metaphor, computing systems, and active learning, *International Journal of Man-Machine Studies*, 22, 39-57.

Cox, R. (1999). Representation construction, externalised cognition and individual differences. *Learning and instruction, Vol. 9*, 343-363.

Cox, R., and Brna, P., (1995). Supporting the Use of External Representations in Problem Solving: The Need for Flexible Learning Environments. *Journal of Artificial Intelligence in Education* 6(2/3), 239-302.

Cox, R., Stenning, K., & Oberlander, J. (1995). The effect of interactive multimedia interfaces upon representation selection. In Lee, J. (Ed.), *Proceedings of the First International Workshop on Intelligence and Multimodality in Multimedia Interfaces: Research and Applications (IMMI-1)*, Edinburgh, July, 1995.

Davis, R. B. (1979). In: Solomon, C. (1986). *Computer Environments for Children: A Reflection on Theories of Learning and Education*, The MIT Press, Cambridge, Massachusetts, London, England.

De Corte, E., and Verschaffel, L. (1985). Beginning First Graders' Initial Representation of Arithmetic Word Problems, *Journal of Mathematical Behavior*, 4, 3-21.

De Corte, E., and Verschaffel, L. (1987). Children's Problem-Solving Skills and Processes With Respect to Elementary Arithmetic Word Problems. In: De Corte, E., Lodewijks, H., Parmentier, R., and Span, P. (Eds.), *Learning and Instruction: European Research in an International Context: Volume 1*, 297-308, Leuven University Press and Pergamon Press, Oxford, New York, Beijing, Frankfurt, Sao Paulo, Sydney, Tokyo, Toronto.

De Corte, E., Verschaffel, L., and Van Coille, V. (1988). Influence of Number Size, Problem Structure, and Response Mode on Children's Solutions of Multiplication Word Problems, *Journal of Mathematical Behavior*, 7, 197-216.

De Corte, E. (1995). Fostering Cognitive Growth: A Perspective From Research on Mathematics Learning and Instruction, *Educational Psychologist*, 30, 1, 37-46.

De Corte, E. (1996). In: Vosniadou, S., De Corte, E., Glaser, R., and Mandl, H., *International Perspectives on the Design of Technology-Supported Learning Environments*, Lawrence Erlbaum Associates, Publishers, Mahwah, New Jersey.

De Corte, E., Greer, B., and Verschaffel, L. (1996). Mathematics Teaching and Learning. In: Berliner, D. C., and Calfee, R. C. (Eds.), *Handbook of Educational Psychology*, Simon and Schuster Macmillan, New York.

Demana, F., and Osborne, A. (1988). Choosing a Calculator: Four-Function Foul-ups, *Arithmetic Teacher*, 35, 7, 2-3.

Department for Education (1995). *Mathematics in the National Curriculum*, HSMO, London.

Derry, S., Hawkes, L. W., Diefenbach, T., and Kegelmann, H. (1993). An experimental tool for developing metacognitive awareness in a problem-solving context. In: Nwana, H. S. (Ed.), *Mathematical Intelligent Learning Environments*, Intellect, Oxford, England.

Dix, A., Finlay, J., Abowd, G., and Beale, R. (1993). *Human-Computer Interaction*, Prentice Hall.

Dörfler, W., Computer Use and Views of the Mind. In: Keitel, C., and Ruthven, K. (Eds.) (1993), Learning from Computers: Mathematics Education and Technology, *NATO ASI Series, Series F: Computer and Systems Sciences*, 121.

Dreyfus, T. (1993). Didactic Design of Computer-based Learning Environments. In: Keitel, C., and Ruthven, K. (Eds.) Learning From Computers: Mathematics Education and Technology, *NATO ASI Series F: Computer and Systems Sciences*, Vol. 121.

Dufour-Janvier, B., Bednarz, N, and Belanger, M. (1987). Pedagogical Considerations Concerning the Problem of Representation. In: Janvier, C. (Ed.), *Problems of Representation in the Teaching and Learning of Mathematics*, Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey, London.

Dumont, B. (1993). A diagnostic knowledge-based system for fractions. In: Nwana, H. S. (Ed.), *Mathematical Intelligent Learning Environments*, Intellect, Oxford, England.

Eberts, R., E., and K. P. Bittianda (1993). Preferred mental models for direct manipulation and command-based interfaces, *Man-Machine Studies*, 38, 769-785.

Ecker, M. W. (1989). Mathematical Computing: Order of Operations and New Software, *Journal of Computers in Mathematics and Science Teaching*, 8, 2, 103-105.

Friedhoff, R. M., and Benzon, W. (1991). *The Second Computer Revolution: Visualization*, W. H. Freeman and Company, New York.

Fuson, K. C. (1992). Research on Whole Number Addition and Subtraction. In: Grouws, D. A. (Ed.), *Handbook of Research on Mathematics Teaching and Learning*, Macmillan Publishing Company, New York.

Glaser, R., Ferguson, E. L., and Vosniadou, S. (1996). Introduction: Cognition and the Design of Environments for Learning. In: Vosniadou, S., De Corte, E., Glaser, R., and

Goldin, G. A. (1987). Cognitive Representational Systems for Mathematical Problem Solving (p. 125-145). In: Janvier, C. (Ed.), Problems of Representation in the Teaching and Learning of Mathematics, Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey, London.

Goldin, G. A. (1992b). Toward an Assessment Framework for School Mathematics. In: Berliner, D. C., and Calfee, R. C. (Eds.), Handbook of Educational Psychology (pp. 502-503), Simon and Schuster Macmillan, New York.

Green, T. R. G. (1989). Cognitive Dimensions of Notations. In: Winder, R., and Sutcliffe. A. (Eds.), *People and Computers V*, Cambridge University Press.

Green, T. R. G., and Petre, M. (1996). Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework, *Journal of Visual Languages and Computing*, 7, 131-174.

Green, T. R. G., and Blackwell, A. (1998). Cognitive Dimensions of Information Artefacts: A Tutorial. *Http://www.ndirect.co.uk/~thomas.green/workStuff/Papers/*

Green, T. R. G. (1999). Building and manipulating complex information structures: Issues in Prolog programming. In: Brna, P., du Boulay, B., and Pain, H. (Eds.), *Learning to Build and Comprehend Complex Information Structures: Prolog as a Case Study* .Stamford, CT: Ablex, 1999, 7-27.

Greeno, J. G. (1983). Conceptual Entities. In: In: Gentner, D, and Stevens, A. L. (Eds.), *Mental Models*, Lawrence Erlbaum Associates, Hillsdale, New Jersey, London.

Greeno, J. G. (1991a). Number Sense as Situated Knowing in a Conceptual Domain, *Journal for Research in Mathematics Education*, 22, 3, 170-218.

Greeno, J. G. (1991b). A View of Mathematical Problem Solving in School. In: Smith, M. U. (Ed.), *Toward a Unified Theory of Problem Solving*, Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey, Hove, London.

Greer, B. (1992). Multiplication and Division as Models of Situations. In: Grouws, D. A. (Ed.), *Handbook of Research on Mathematics Teaching and Learning*, Macmillan Publishing Company, New York.

Grossen, G., and Carnine, D., (1990). Diagramming a Logical Strategy: Effects on Difficult Problem Types and Transfer. *Learning Disability Quarterly* 13, 168-182.

Halasz, F. G., and Moran, T. P. (1983). Mental Models and Problem Solving in Using a Calculator, *CHI 1983 Conference Proceedings: Human Factors in Computing Systems*, 212-216.

Hawkins, V. J. (1984). Sharing Teaching Ideas, *Mathematics Teacher*, 77, 6, 448-451, 490.

Heid, M. K. (1990). How Symbolic Mathematical Systems Could and Should Affect Precollege Mathematics, *Mathematics Teacher*, 83, 9, 716-720.

Hembree, R., and Dessart, D. J. (1986). Effects of Hand-Held Calculators in Precollege Mathematics Education: A Meta-Analysis, *Journal for Research in Mathematics Education*, 17, 2, 83-99.

Hembree, R., and Dessart, D. (1992). Research on Calculators in Mathematics Education. In: Fey, J., and Hirsch. C. (Eds.). *Calculators in Mathematics Education*, Reston VA: National Council of Teachers of Mathematics.

Hiebert, J. (1988). A Theory of Developing Competence With Written Mathematical Symbols, *Educational Studies in Mathematics*, 19, 333-355.

Hiebert, J., and Carpenter, T. P. (1992). Learning and Teaching With Understanding. In: Grouws, D. A. (Ed.), *Handbook of Research on Mathematics Teaching and Learning*, Macmillan Publishing Company.

Hiebert, J., and Lefevre, P. (1986). Conceptual and Procedural Knowledge in Mathematics: An Introductory Analysis. In: Hiebert, J. (Ed.), *Conceptual and Procedural Knowledge: The Case of Mathematics*, Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey, London.

Hoyles, C., Sutherland, R., and Healy, L. (1991). In: Sivasubramaniam, P. Distributed Cognition, Computers and the Interpretation of Graphs, *Proceedings of The British Society for Research into Learning Mathematics*, 59-64, Leeds University, 14 November, 1998.

Janvier, C. (1987). Presentation and Understanding: The Notion of Function as an Example. In: Janvier, C. (Ed.), *Problems of Representation in the Teaching and Learning of Mathematics*, Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey, London.

Johnson-Laird, P. N. (1983). *Mental Models*. Cambridge: Cambridge University Press.

Kahn, K. (1996). An Animated Programming Environment for Children, *Journal of Visual Languages and Computing*, 7, 197-217.

Kaptelinin, V. (1992). Integration of Computer Tools Into the Structure of Human Activity: Implications for Cognitive Ergonomics. In: Van der Veer, G. C., Tauber, M. J., Kaput, J. J. (1987). Toward a Theory of Symbol use in Mathematics. In: Janvier, C. (Ed.), *Problems of Representation in the Teaching and Learning of Mathematics* (pp. 159-195), Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey, London.

Kaput, J. J. (1989). Supporting Concrete Visual Thinking in Multiplicative Reasoning: Difficulties and Opportunities, *Focus on Learning Problems in Mathematics*, 11, 1, 35-47.

Kaput, J. J. (1992). Technology and Mathematics Education. In: Grouws, D. A. (Ed.), *Handbook of Research on Mathematics Teaching and Learning*, Macmillan Publishing Company, New York.

Bagnara, S., and Antalovits, M. (Eds.), Human Computer Interaction: Tasks and Organisation, *Proceedings of the 6th European Conference on Cognitive Ergonomics (ECCE 6)*, Balatonfured, Hungary.

Kozma, R. B., Russell, J., Jones, T., Marx, N., and Davies, J. (1996). The Use of Multiple, Linked Representations to Facilitate Science Understanding. In: Vosniadou, S., De Corte, E., Glaser, R., and Mandl, H., *International Perspectives on the Design of Technology-Supported Learning Environments*, Lawrence Erlbaum Associates, Publishers, Mahwah, New Jersey.

Larkin, J. H., and Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words, *Cognitive Science*, 11, 65-99.

LeBlanc, M. D. (1993). In: Nwana, H. S. (Ed.), *Mathematical Intelligent Learning Environments*, Intellect, Oxford, England.

Lehtenin, E., and Repo, S. (1996). Activity, Social Interaction, and Reflective Abstraction: Learning Advanced Mathematical Concepts in a Computer Environment. In: Vosniadou, S., De Corte, E., Glaser, R., and Mandl, H. *International Perspectives on the Design of Technology-Supported Learning Environments*, LEA Publishers.

Lesh, R. (1987). In: Kaput, J. J. (1992). Technology and Mathematics Education. In: Grouws, D. A. (Ed.), *Handbook of Research on Mathematics Teaching and Learning*, Macmillan Publishing Company, New York.

Lesh, R., Post, T., and Behr, M. (1987). Representations and Translations Among Representations in Mathematics Learning and Problem Solving. In: Janvier, C. (Ed.), Problems of Representation in the Teaching and Learning of Mathematics, Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey, London.

Lowe, R. (1993). *Successful Instructional Diagrams*, Kogan Page, London, Philadelphia, 47-63.

Lowe, R.K. (1994). Selectivity in Diagrams; Reading Beyond the Lines. *Educational Psychology* 14, 467-491.

Mandl, H. (Eds.), *International Perspectives on the Design of Technolpogy-Supported Learning Environments*, Lawrence Erlbaum Associates, Publishersd, Mahwah, New Jersey.w

Marton, F. and Säljö, R. (1984). In: Vermunt, J. D. (1998). The Regulation of Constructive Learning Processes, *British Journal of Educational Psychology*, 68, 149-171.

Mathsoft, Inc. (1995). *MathCad* [Software].

Mayer, R. E., and Bayman, P. (1981). Psychology of Calculator Languages: A Framework for Describing Differences in User's Knowledge, *Communications of the ACM*, 24, 8, 511-520.

Mayer, R. E. And Wittrock, M. C. (1996). Problem-Solving Transfer. In: Berliner, D. C., and Calfee, R. C. (Eds.), *Handbook of Educational Psychology*, Simon and Schuster

Macmillan, New York. Prentice Hall International, London, Mexico City, New Delhi, Singapore, Sydney, Toronto.

Nardi, B. A. (1996). Activity Theory and Human-Computer Interaction. In: Nardi, B. A. (Ed.), *Context and Consciousness: Activity Theory and Human-Computer Interaction*, The MIT Press, Cambridge, Massachusetts, London, England.

Nathan, M. J., and Resnick, L. B. (1993). Less Can Be More: Unintelligent Tutoring Based on Psychological Theories and Experimentation, *Technology Based Environments*, 183-192.

National Council of Teachers of Mathematics (NCTM) (1974). In: Hembree, R., and Dessart, D. J. (1986). Effects of Hand-Held Calculators in Precollege Mathematics Education: A Meta-Analysis, *Journal for Research in Mathematics Education*, 17, 83-99.

Nesher, P. In: Greer, B., Multiplication and Division as Models of Situations. In: Grouws, D. A. (Ed.) (1992), *Handbook of Research on Mathematics Teaching and Learning*, Macmillan Publishing Company, New York.

Nesher , P. (1988). In: Hiebert, J. and Behr, M. (Eds.). *Number Concepts and Operations in the Middle Grades, Volume 2*, Lawrence Erlbaum Associates.

Nwana, H. S. (1993). The anatomy of FITS: A Mathematic tutor. In: Nwana, H. S. (Ed.), *Mathematical Intelligent Learning Environments*, Intellect, Oxford, England.

Ohlsson, S. (1987). Sense and Reference in the Design of Interactive Illustrations for Rational Numbers. In: Lawler, R. W., and Yazdani, M. (Eds.), *Artificial Intelligence and Education Volume one: learning environments and tutoring systems*, Ablex Publishing, New Jersey.

O'Reilly, D. (1999). Students' Interpretations and Representations of Number. In: Brown, L. (Ed.), Making Meanings in Mathematics: Visions of Mathematics No. 2, Advances in Mathematics Education No. 1 (A collection of extended and refereed papers from BSRLM - The British Society for Research into Learning Mathematics. QED and BSRLM.

Paivio, A. (1986). *Mental Representations: A Dual-Coding Approach*, Oxford University Press, Oxford.

Peled, I., and Resnick, L. B. (1988). In: Schwarz, B. B., Nathan, M., and Resnick, L. B. (1996). Acquisition of Meaning for Arithmetic Structures With the Planner. In: Vosniadou, S., De Corte, E., Glaser, R., and Mandl, H., *International Perspectives on the Design of Technology-Supported Learning Environments*, Lawrence Erlbaum Associates, Publishers, Mahwah, New Jersey.

Polson, P. G., Lewis, C., Rieman, J., and Wharton, C. (1992). Cognitive walkthroughs: a method for theory-based evaluation of user interfaces, *International Journal of Man-Machine Studies*, 36, 741-773.

Repenning, A., Ioannidou, A, and Ambach, J. (1998). Learn to Communicate and Communicate to Learn: How constructing educational software components that can be shared and linked together can be a viable and effective educational experience, (pp. 1-28), *Submission to JIME.*

Riel, M. M., Levin, J. A., and Miller-Souviney, B. (1987). Learning With Interactive Media: Dynamic Support for Students and Teachers. In: Lawler, R. W., and Yazdani,

M. (Eds.), *Artificial Intelligence and Education Volume one: learning environments and tutoring systems*, Ablex Publishing, New Jersey.

Rieman, J., Franzke, M., and Redmiles, D. (1995). Usability Evaluation with the CognitiveWalkthrough.*Http://www.acm.org/sigchi/chi95/proceedings/tutors/jr_bdy.htm*

Ruthven, K., and Chaplin, D. (1997). The Calculator as a Cognitive Tool: Upper-Primary Pupils Tackling a Realistic Number Problem. *International Journal of Computers for Mathematical Learning*, 2, 93-104.

Simons, P. R. J. (1993). Constructive Learning: The Role of the Learner. In: Duffy, T., Lowyck, T., and Jonassen, D. (Eds.), *Designing Environments for Constructive Learning*, Berlin Springer: Verlag, 291-313.

Salomon, G. (1992). Effects with and of computers and the study of computer-based learning environments. In: De Corte, E., Linn, M. C., Mandl, H., and Verschaffel, L. (Eds.), Computer-Based Learning Environments and Problem Solving, *NATO ASI Series, Series F: Computer and Systems Sciences*, 84.

Schär, S. G. (1996). The Influence of the user interface on solving well- and ill-defined problems, *International Journal of Human-Computer Studies*, 44 1-18.

Schnotz, W., Picard, E., & Hron, A. (1993). How Do Successful and Unsuccessful Learners Use Text and Graphics? *Learning and Instruction* 20 (3), 181-199.

Schwarz, J. (1982). The Semantic Calculator: Solving the Word-Problem Problem, *Classroom Computer News*, 2, 22-24.

Schwarz, B. B., Nathan, M., and Resnick, L. B. (1996). Acquisition of Meaning for Arithmetic Structures With the Planner. In: Vosniadou, S., De Corte, E., Glaser, R., and Mandl, H., *International Perspectives on the Design of Technology-Supported Learning Environments*, Lawrence Erlbaum Associates, Publishers, Mahwah, New Jersey.

Shneiderman, B. (1987). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison-Wesley, USA.

Shuard, H., Walsh, A., Goodwin, J., and Worcester, V. (1991). *Calculators, Children and Mathematics*, Simon and Schuster, London.

Shuell, T. J. (1992). Designing Instructional Computing Systems (ICSs) for Meaningful Learning, *NATO ASI Series, Series F: Computer and Systems Sciences*, 85, pp. 19-54, Springer-Verlag.

Shuell, T. J. (1996). Teaching and Learning in a Classroom Context. In: Berliner, D. C., and Calfee, R. C. (Eds.), *Handbook of Educational Psychology* (pp. 726-764), Simon and Schuster Macmillan, New York.

Sleeman, D. H. (1983). Inferring Student Models for Intelligent Computer-aided Instruction. In: Michalski, R . S ., Carbonell, J. G., and Mitchell, T. M. (Eds.), *Machine Learning: An Artificial Intelligence Approach*, Springer-Verlag, Berlin, Heidelberg, New York, Tokyo.

Solomon, C. (1986). *Computer Environments for Children: A Reflection on Theories of Learning and Education*, The MIT Press, Cambridge, Massachusetts, London, England.

Somekh, B., (1996). Designing Software to Maximize Learning. In: *Association for Learning Technology Journal* Volume 4 Number 3, 4-16.

Stary, C., and Peschl, M. F. (1998). Representation still matters: cognitive engineering and user interface design. *Behaviour and Information Technology*, 17, 6, 338-360.

Steffe, L. P. (1988). Children's Construction of Number Sequences and Multiplying Schemes. In: Hiebert, J., and Behr, M. (Eds.), *Number Concepts and Operations in the Middle Grades*, Volume 2.

Tait, K., Hartley, J. R., and Anderson, R. C. (1973). Feedback Procedures in Computer-assisted Arithmetic Instruction, *The British Journal of Educational Psychology*, 43, 1, 161-171.

Thimbleby, H. (1995). A New Calculator and Why it is Necessary, The Computer Journal, 38, 6, 418-433.

Thompson, P. W., and Thompson, A. G. (1990). Salient Aspect s of Experiences With Concrete Manipulatives, *Proceedings: 14th Conference of the International Group for the Psychology of Mathematics Education*, 337-343.

Vermunt, J. D. (1998). The Regulation of Constructive Learning Processes, *British Journal of Educational Psychology*, 68, 149-171.

Van Labeke, N., and Ainsworth, S. (2001). Applying the DeFT Framework to the Design of Multi-Representational Instructional Simulations,

Vivet, M., Delozanne, E., and Carriere, E. (1988). In: Delozanne, E., Bruillard, E., and Vivet, M., Mathematical learning environments: A French viewpoint (p. 15). In: Nwana, H. S. (Ed.), *Mathematical Intelligent Learning Environments*, Intellect, Oxford, England.

Vygotsky, L. (1978). *Mind in Society*. Cambridge, MA: Harvard University Press.

Vygotsky, L., (1986). *Thought and Language*, Cambridge MA: MIT Press.

Wiebe, J. H. (1989). Teaching Mathematics With Technology, *Arithmetic Teacher*, 37, 3, 36-38.

Wiebe, J. H. (1989). Calculator Memory and Multistep Problems, *Arithmetic Teacher*, 37, 1, 48-49.

Williams, E., and Shuard, H. (1994). *Primary Mathematics Today: Fourth Edition*, Longman.

Woods, P., and Hartley, J. R. (1971). Some Learning Models for Arithmetic Tasks and Their use in Computer Based Learning. In: Nisbet, J. D. (Ed.), *The British Journal of Educational Psychology*, 41, 38-48.

Yazdani, M. (1987). Artificial Intelligence, Powerful Ideas and Children's Learning. In: Rutkowska, J. C. and Crook, C. (Eds.), *Computers, Cognition and Development: Issues for Psychology and Education*, John Wiley and Sons, Chichester, New York, Brisbane, Toronto, Singapore.

Yvon, B. R. (1987). A Compelling Case for Calculators, *Arithmetic Teacher*, 34, 6, 16-19.

Ziegler, J. E. and Faehnrich, K. P. In: Eberts, R., E., and K. P. Bittianda (1993). Preferred mental models for direct manipulation and command-based interfaces, *Man-Machine Studies*, 38, 769-785.

Zhang, J. (1997). The Nature of External Representations in Problem Solving, *Cognitive Science*, 21, 179-217

APPENDICES

Appendix 1

The Cognitive Walkthrough

The extract below was taken from the internet and provides relevant information regarding cognitive walkthroughs.

Theory Underlying The Walkthrough

Rieman, et.al (1995) state the following. "The cognitive walkthrough is a practical evaluation technique grounded in Lewis and Polson's CE+ theory of exploratory learning [3,4,5]. The CE+ theory is an information-processing model of human cognition that describes human-computer interaction in terms four steps:

1) The user sets a goal to be accomplished with the system (for example, "check spelling of this document").

2) The user searches the interface for currently available actions (menu items, buttons, command-line inputs, etc.).

3) The user selects the action that seems likely to make progress toward the goal.

4) The user performs the selected action and evaluates the system's feedback for evidence that progress is being made toward the current goal.

For most realistic tasks that a user would attempt with a system, these four steps are repeated many times to achieve a series of subgoals that define the complete task. The cognitive walkthrough examines each of the correct actions needed to accomplish a task, and evaluates whether the four cognitive steps will accurately lead to those actions.

The Walkthrough Procedure

Prerequisites to the walkthrough include: (1) a general description of who the users will be and what relevant knowledge they possess, (2) a specific description of one or more representative tasks to be performed with the system, and (3) a list of the correct actions required to complete each of these tasks with the interface being evaluated.

The walkthrough is typically performed by the interface designer and a group of his or her peers. Small-scale walkthroughs of parts of an interface can also be done by individual designers as they consider alternative designs. In a group situation, one of the evaluators usually takes on the duties of "scribe," recording the results of the evaluation as it proceeds, and another group member acts as facilitator, to keep the evaluation moving.

With the prerequisites assembled and duties assigned, the walkthrough process involves examining each individual step in the correct action sequence and trying to tell a believable story about why the prospective user would choose that action. Note that this is not an open forum approach of predicting what activities the user might engage in, given this interface and task. It is specifically limited to considering whether the user will select each of the correct actions along the solution path.

In many cases, the group of evaluators will readily agree that the user will select the correct action, and no further analysis is required. For example, the first action in using a Macintosh program may be to double-click its icon; the evaluators could readily agree that experienced Mac users would have little trouble with this step. Other cases, however, may be less clear. To assess the ease with which the correct action will be selected, the walkthrough process suggests four criteria for evaluating the stories told about the users' actions.

The four criteria for evaluating the stories directly reflect the information-processing model that underlies the walkthrough. They ask the evaluators to consider the user's goal, the accessibility of the of the correct control, the quality of the match between the control's label and the goal, and the feedback provided after the control is acted on" (pp. 2-3).

Appendix 2

Cognitive Dimensions

Cognitive Dimensions

Cognitive dimensions is a method designed to help evaluate the usability of systems which store, manipulate and display information (Green and Blackwell, 1998). The systems may be either interactive (e.g. word-processors, software environments, mobile telephones), or non-interactive (e.g. tables, graphs, programming languages). The cognitive dimensions approach uses a set of terms which serve as a checklist to ensure that potential problems or improvements in usability (e.g. software design) are not overlooked.

The *dimensions* are actually a list of fourteen descriptors which help focus attention on usability issues, and thus may be used to provide a means by which design alternatives can be compared (Green, 1999). The dimensions are listed below in Table A1.

Table A1.1 The Dimensions

Abstraction (usability difficulties)	Mechanisms which result in abstraction during the usability of an artefact.
Closeness of mapping	Mapping between the problem world (domain) and an artefact's usability (i.e. its representations) should be as close as

	possible.
Consistency	The semantics of text are expressed using similar grammatical forms.
Diffuseness (i.e. spread out)	Using tautologous, redundant, or over wordiness in language.
Error-proneness	The use of notation tends to lead to mistakes.
Hard mental operations	There is a high demand on cognitive resources.
Hidden dependencies	A relationship between two components (e.g. text and graphics) at an interface, where one component which is dependent on the other is not apparent but hidden.
Premature Commitment	A user will make a premature commitment at an interface when he/she is forced to make a decision before a required piece of information is available.
Progressive Evaluation	A system enables work to be checked at any time.
Provisionality	A user has to meet certain requirements (provisions) in order to carry out actions at an interface. Provisionality refers to the degree of commitment a user makes to actions.

Role-expressiveness	The ease with which syntax is understood such that: (a) notation can be split into its components parts; and (b) the relationships between the components can be identified (e.g. Prolog programs make little use of role-expressiveness).
Secondary notation	Extra notation (e.g. circuit diagrams used by engineers) is provided in order to supplement role –expressiveness.
Viscosity	This is resistance to change. In other words, it is the amount of work a person has to input at an interface in order to bring about a small change in the system's behaviour. Thus, the lower the viscosity, the easier it is for a person to interact.
Visibility	Does a system enable its component parts (e.g. Prolog programs) to be viewed easily.

Since the dimensions are not themselves evaluative, two other components within the cognitive dimensions framework are used. These are *activity types* and *environment.*

Four types of activity

1. *Incrementation*: adding information (e.g. adding a formula to a spreadsheet).

2. *Transcription*: copying or converting information (e.g. converting a formula into a spreadsheet).

3. *Modification*: changing or modifying information (e.g. changing the layout of a spreadsheet or modifying it to compute a different problem).

4. *Exploratory design*: where a final product cannot be envisaged, but is formulated through sketches and other means.

The Environment

Green (1999) points out that cognitive dimensions are not only properties of the *notation* used (e.g. as with Prolog), but they are also properties of the *environment* (e.g. as with a designed computer-based learning or Prolog environment). Thus, a particular notation may be viscous (i.e. hard to change) where it is simply part of many lines of programming code, yet the same notation may be quite fluid when used in an environment which has appropriate editing tools.

Final Comments

Green (1999) states that despite cognitive dimensions offering a useful framework for analysis, the framework needs further development. In particular, with regard to Prolog, computer programmers at all levels of experience could be helped by having an editing enviroment which manages abstractions thereby reducing the viscosity of Prolog. However, this is not easy but is a way forward for improving the cognitive dimensions system.

Appendix 3

The Pilot Test Questions

Please attempt to answer the following questions using the computer based learning system ENCAL.

1. Jill has two red books and three blue books on one shelf in a bookcase.

Iconic Representation

(a) Use the book, shelf, and bookcase icons to represent the items in Jill's room.

(b) Count how many books Jill has in her room altogether. Write down your answer.

....

[CLEAR SCREEN WHEN FINISHED]

2. John places three red books on one shelf in one of the bookcases. He later adds three blue books to the shelf.

Iconic Representation

(a) Use the book, shelf, and bookcase icons to represent the items in John's room.

[CLEAR SCREEN WHEN FINISHED]

Dataflow Representation

(b) Construct a dataflow tree which represents the items in John's room.

(c) Write down the arithmetic expression from the dataflow tree which represents the number of books in John's room.

....

[CLEAR SCREEN WHEN FINISHED]

3. Jill has three red books on each of five shelves in one bookcase.

Iconic Representation

(a) Use the book, shelf, and bookcase icons to represent the items in Jill's room.

[CLEAR SCREEN WHEN FINISHED]

Dataflow Representation

(b) Construct a dataflow tree to represent the items in

Jill's room.

(c) Write down the arithmetic expression from the dataflow tree.

....

(d) Check that the dataflow tree numbers match with the book and shelf icons.

[CLEAR SCREEN WHEN FINISHED]

Calculator Representation

(e) Use the calculator to determine how many books there are altogether in Jill's room.

Check that the numbers match the book and shelf icons. Write down your answer.

....

[CLEAR SCREEN WHEN FINISHED]

4. John puts four blue books on each of two shelves in one bookcase. He then puts two blue books on each of three shelves in another bookcase.

Iconic Representation

(a) Represent the items in John's room using the book, shelf, and bookcase icons.

[CLEAR SCREEN WHEN FINISHED]

Dataflow Representation

(b) Construct a dataflow tree to represent the items in John's room.

(c) Write down the arithmetic expression from the dataflow.

....

[CLEAR SCREEN WHEN FINISHED]

Calculator Representation

(d) Use the calculator to determine how many books there are altogether in John's room.

Write down your answer.

....

(e) Check that the calculator answer matches the total number of books in the iconic representation.

[CLEAR SCREEN WHEN FINISHED]

Dataflow Representation

(f) Reconstruct the dataflow tree which represents the items in John's room.

(g) Place a rectangle around 2x4 and 3x2 by selecting [] from the menu.

(h) Write down the arithmetic expression from the dataflow. Use brackets to represent each rectangle.

....

Calculator Representation

(i) Use the calculator again to determine how many books there are altogether in John's room. Write down your answer.

....

(j) Which answer do you think is correct, (d) or (h)? Write (d) or (h) below.

....

THANK YOU FOR YOUR HELP

Appendix 4

Questions for the Evaluation of ENCAL

Background Information

The aim of ENCAL - a computer-based learning system, is to help you develop your concept of number and your multiplication and addition skills with the help of an on-screen calculator and some additional computer-based support.

Translating a written problem to a mathematical calculation is hard, in particular the use of brackets; and the way in which calculators behave is not clear. So how can we help you with this?

The ENCAL program helps to show what the calculator is doing. Let's see if it works! Try the problems.

HOW TO USE THE SYSTEM: SOME DEMONSTRATIONS

Q1 Place three blue books on each of five shelves in one bookcase.

(a) Count how many books there are in the bookcase.

Write down the answer……………………………………………

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

(b) Use the calculator and the information in question 1 to find how many books there are in the bookcase.

Write down the answer……………………………………………

Double check this answer by counting the number of books in the bookcase.

Are both answers the same, yes/no?……………………………………………

Click on "look inside" on the calculator and write down the contents below

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

Q2 Place two shelves in one bookcase, and put four blue books and two green books on each shelf.

(a) Count how many books there are in the bookcase.

Write down the answer..

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

(b) Use the calculator and the information in question 2 to find how many books there are in the bookcase.

Write down the answer...

Double check this answer by counting the number of books in the bookcase.

Are both answers the same, yes/no?...

Click on "look inside" on the calculator and write down the contents below

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

EXIT AND RESTART ENCAL

HOW TO USE THE SYSTEM: SOME DEMONSTRATIONS

Q1 Place five shelves in one bookcase, and on each shelf place three books.

(a) Count how many books there are in the bookcase

Write down the answer...

(b) Using the boxes on the top half of the screen to help you, write down the equation that represents the problem in question 1 (the dotted rectangles represent the brackets normally used in maths).

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

(c) Use the calculator and the information in question 1 to find how many books there are in the bookcase.

Write down the answer...

Double check this answer by counting t he number of books in the bookcase.

Are both answers the same, yes/no?..

Click on "look inside" on the calculator and write down the contents below

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

Q2 Place two shelves in one bookcase, then on each shelf place four blue books and two green books.

(a) Count how many books there are in the bookcase

Write down the answer...

(b) Using the boxes on the top half of the screen to help you, write down the equation that represents the problem in question 2 (the dotted rectangles represent the brackets normally used in maths).

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

(c) Use the calculator and the information in question 2 to find how many books there are in the bookcase.

Write down the answer...

Double check this answer by counting the number of books in the bookcase.

Are both answers the same, yes/no?..

Click on "look inside" on the calculator and write down the contents below

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

EXIT AND RESTART ENCAL

THE TEST QUESTIONS

GROUP 1

Q1 Place one shelf in a bookcase, then on it place two green books and three blue books.

(a) Count how many books there are in the bookcase.

Write down the answer..

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

(b) Use the calculator and the information in question 2 to find how many books there are in the bookcase.

Write down the answer...

Double check this answer by counting the number of books in the bookcase.

Are both answers the same, yes/no?..

Click on "look inside" on the calculator and write down the contents below

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

Q2 Place one shelf in one bookcase, and two shelves in another bookcase. On the first shelf place three green books and three blue books, and on the second two shelves place two blue books.

(a) Count how many books there are in the bookcase.

Write down the answer...

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

(b) Use the calculator and the information in question 2 to find how many books there are in the bookcase.

Write down the answer...

Click on "look inside" on the calculator and write down the contents below

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

Q3 Place two shelves in one bookcase, then place four blue books and two green books on each shelf. In another bookcase place three shelves, then place three blue books and two green books on each of these shelves.

(a) Count how many books there are in the bookcase

Write down the answer..

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

(b) Use the calculator and the information in question 1 to find how many books there are in the bookcase.

Write down the answer..

Click on "look inside" on the calculator and write down the contents below

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

EXIT AND RESTART ENCAL

THE TEST QUESTIONS

GROUP 2

Q1 Place one shelf in a bookcase, and on it place two green books and three blue books.

(a) Count how many books there are in the bookcase

Write down the answer...

(b) Using the boxes on the top half of the screen to help you, write down the equation that represents the problem in question 1 (the dotted rectangles represent the brackets normally used in maths).

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

(c) Use the calculator and the information in question 2 to find how many books there are in the bookcase.

Write down the answer...

Click on "look inside" on the calculator and write down the contents below

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

Q2 Place one shelf in a bookcase and on it place three green books and three blue books. Place two shelves in another bookcase, and on each of these shelves place two blue books.

(a) Count how many books there are in the bookcase

Write down the answer..

(b) Using the connected rectangles on the top half of the screen to help you, write down the equation that represents the problem in question 2 (the rectangles represent brackets).

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

(c) Use the calculator and the information in question 2 to find how many books there are in the bookcase.

Write down the answer..

Click on "look inside" on the calculator and write down the contents below

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

Q3 Place two shelves in one bookcase, and on it place four blue books and two green books. Then place three shelves in another bookcase, and on each of these shelves place three blue books and two green books.

(a) Count how many books there are in the bookcase

Write down the answer...

(b) Using the boxes on the top half of the screen to help you, write down the equation that represents the problem in question 3 (the dotted rectangles represent the brackets normally used in maths).

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

(c) Use the calculator and the information in question 3 to find how many books there are in the bookcase.

Write down the answer...

Click on "look inside" on the calculator and write down the contents below

PRESS THE CA BUTTON ON THE CALCULATOR TO CLEAR THE SCREEN

THANK YOU FOR YOUR PARTICIPATION

Appendix 5

Helpers' Notes During the Evaluation of ENCAL

The general mathematical aims and objectives of ENCAL together include helping: (a) evaluation sequences, (b) the correct use of parentheses, and (c) appropriate mental model development for calculator use. Participants should use the L – R calculator in order to: appreciate the inadequacies of calculator behaviour; and gain maximum benefit from the datatree.

ENCAL consists of three representations: iconic, datatree, and calculator. The words *iconic* and *datatree* may be confusing to participants. Consequently, the following words: books, shelves, bookcases (icons), and connected rectangles (datatree) should be used.

The datatree is a crucial aspect of ENCAL, because this provides intermediary help during problem solving. You'll see that there are demonstration questions followed by the test questions. Group1 will not be trained in the use of the datatree, so group 2 only will be prompted to use the datatree in the test questions. The aim is to compare the performance of the two groups.

The demonstration questions are to be directive (i.e. incremental). In other words, pupils should be directed and shown:

- how to use a specific representation;
- how to use representations during the stages of problem solving;
- which representations to use and the appropriate order of use during problem solving.

The test questions are to be less directive (i.e. open-ended). In other words, pupils should **not** be directed or shown, but may be reminded if necessary:

- how to use a specific representation;
- how to use representations during the stages of problem solving;
- which representations to use and the appropriate order of use during problem solving.

A video camera will be used to record each participant's interaction at the computer. The helper who sits with participants at the computer will have to switch the camera on/off at the start/end of each participant's turn. In addition to the helper sat beside a participant, another helper should be sat behind taking notes of each participant's actions. The notes should include:

- the name, age, and sex of each participant;
- whether group 1 or group 2, and whether the demonstration or the test questions;
- which representations were used when during the solving of each problem;
- what a participant did at the computer when difficulties were encountered.

Appendix 6

Examples of the Coding Scheme When Classifying Behaviours

Operator Problem (OP)

Example 1

Question 2(b). Participant R. L. answered this incorrectly. That is, he entered 2 + 26 into the calculator. He missed out the operator between the 2 and the 6. This was then classified as an operator problem (OP).

Example 2

Question 3(b). Participant D. P. used the addition operator only, instead of combining addition and multiplication. He entered 4 + 4 + 2 + 2 instead of 2 x 4 + 2. Therefore, this was classified as an operator problem (OP).

Brackets Problem (BP)

Example 3

Question 3(a). Participant R. W. was confused by the use of nested brackets, and she obtained the incorrect answer. As she evaluated the expression (2 x (4 + 2)) + (3 x (3 + 2)) she ended up with (12 + 3) x 5 = 15 x 5 = 75. This was classified as a brackets problem (BP).

Understanding Problem (UP)

Example 4

Question 2(b). Participant R. L. became confused and appeared to invent working. Therefore, this was classified as understanding the problem (UP).

Mis-Read Calculator (MRC)

Example 5

Question 2(b). Participant D. F. obtained the correct number of books on the shelves (i.e. 10), however he mis-read the calculator and wrote down the answer of 70. This was classified as mis-read calculator (MRC).

Incomplete Calculator Use (ICU)

Example 6

Question 3(b). Participant D. F. only partially finished the question using the calculator. Thus, instead of computing the mathematical expressions for both bookcases, he only did one bookcase. Thus, the problem was classified as incomplete calculator use (ICU).

Wrong Thinking (WT)

Example 7

Question 1(b). Participant C. S. achieved the correct number of books, however instead of having 2 + 3 for his calculator entry, he had 1 + 4. This was therefore classified as wrong thinking (WT).

Wrong Number of Books (WNB)

Example 8

Question 1(a). Participant L. A. wrote down the answer of 6 books despite placing 5 books on the shelf. This was classified as the wrong number of books (WNB).

Question Correct (QCT)

Example 9

Question 3(a, b, c). Participant N. J. correctly answered each part of question 3. Consequently, this was classified as question correct (QCT).

Fragments of Working (FW)

Example 10

Question 2(b). Participant H. P. failed to obtain all the necessary information from the datatree. Thus, only fragments of working were visible (i.e. only parts of the mathematical expression were written down). Hence, this was classified as fragments of working (FW).

Mis-Read Datatree (MRD)

Example 11

Question 3(b). Participant L. D. wrote down the wrong mathematical expression despite having the datatree on the screen. In particular, the brackets (i.e. the dotted rectangles) were read incorrectly. This was classified as mis-read datatree (MRD).

Interpreting Datatree (ID)

Example 12

Question 3(b). Participant E. B incorrectly translated the dotted rectangle information from the datatree to the mathematical expression. For example, she wrote (4 + 2) (2x) + (x3) + 2 instead of (2 x (4 + 2)) + (3 x (3 + 2)). This was classified as interpreting datatree (ID).

Made-Up Working (MUW)

Example 13

Question 3(c). Participant C. S. simply wrote down his own working in answer to this part of the question (i.e. 2 x 11 = 22). This was obviously made-up, and so was classified as made-up working (MUW).

Mis-Read Information (MRI)

Example 14

Question 1(b). Participant D. P. obtained the correct answer of 5, but mis-read the calculation steps from the calculator representation, using the "look-inside" facility. He thus wrote 3 + 5 instead of 3 + 2. This was classified as mis-read information (MRI).

Question Not Answered (QNA)

Example 15

Questions 1(a), 2(a), and 3(a). Participant L. A. failed to address these problem statements, and in each case wrote down the wrong number of books. This was classified as question not answered (QNA).

Slip (*)

Example 16

Question 2(a). Participant E. B. obtained the incorrect answer only because of a slip with a number. That is, she incorrectly read the total number of books in the bookcases as 9 instead of 10.

Appendix 7

Summary of Results From the ENCAL Evaluation

Table A6.1. Summary of Major Performance Errors

	Major Performance Errors		
	Operator Errors	Parentheses Errors	Understanding Word Problems
Datatree Group	0	1	0
No Datatree Group	9	10	3

Table A6.2. Summary of Overall Correct Calculator Answers

	Correct Calculator Answers	Incorrect Calculator Answers
Datatree Group	14	4
No Datatree Group	9	9